



DEICBIAN: UNA DISTRIBUCIÓ GNU/LINUX PERSONALITZADA PER LABORATORIS DE DOCÈNCIA

Memòria del projecte de final de carrera corresponent
als estudis d'Enginyeria Superior en Informàtica pre-
sentat per Antonio Rota Moreno i dirigit per Sergi Ro-
bles Martínez.

Bellaterra, Juny de 2010

El firmant, Sergi Robles Martínez , professor del Departament d'Enginyeria de la Informació i de les Comunicacions de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha sigut realitzada sota la seva direcció per Antoni Rota Moreno

Bellaterra, Juny de 2010

Firmat: Antoni Rota Moreno

A Andreu Greipel i els seus seguidors.

Agraïments

Tinc el plaer i l'obligació d'agrair aquest projecte a en Sergi Robles i en J. Ignacio Toledo per la gran ajuda proporcionada, gràcies. He d'agrair també la col·laboració de tots els membres del departament d'enginyeria de la informació i les comunicacions. Agraïxo també als meus companys i amics la cooperació durant la fase de test del projecte: Adri, Crisis, David, Néstor, Rafa's, Rubén, gràcies. No voldria oblidar a ningú, ni familiars, ni amics ni qualsevol persona que hagi compartit un mínim fragment de vida amb mi. Kafka va escriure: "A partir de cert punt en endavant no hi ha tornada. Aquest és el punt al que s'ha d'arribar.", així que gràcies a tothom per ajudar-me a arribar a aquest punt. Simplement, gràcies.

Índex

1	Introducció	1
1.1	Motivacions	1
1.2	Objectius	2
1.3	Contingut de la memòria	3
2	GNU, Linux i eines de desenvolupament	5
2.1	GNU/Linux	5
2.1.1	GNU	5
2.1.2	Linux	6
2.1.3	Procés d'arrancada	7
2.1.4	El sistema d'arxius	9
2.2	Eines per al desenvolupament	11
2.2.1	Simple-cdd (Custom Debian Distribution)	11
2.2.2	Live Helper	13
2.2.3	Zenity	14
2.2.4	Màquines virtuals	15
3	Anàlisi de requisits	17
3.1	Introducció	17
3.2	Identificació d'usuaris i participants	18
3.3	Catàleg de requisits del sistema	18
3.3.1	Objectius i abast del sistema	18
3.3.2	Descripció general del sistema	19
3.3.3	Requisits funcionals	19
3.3.4	Requisits no funcionals	22

3.4	Resum	23
3.5	Estudi de la viabilitat del projecte	24
4	Disseny i desenvolupament	25
4.1	Metodologia de desenvolupament	25
4.2	Disseny	26
4.3	La distribució: dEICbian	28
4.3.1	De l'iteració base a la generació de l'instal·lador dEICbian	29
4.3.2	Generació de Live dEICbian	33
4.3.3	Generació de dEICbian i del generador de distribucions dEICbian	36
4.3.4	Realització de documentació d'ajuda	41
4.3.5	Interfície i ús del generador	42
5	Planificació i costos	45
5.1	Tasques i planificació temporal	45
5.2	Costos	47
5.3	Resum	48
6	Conclusions	51
6.1	Valoració	51
6.2	Resultats	52
6.3	Línies de continuïtat	54
	Bibliografia	57

Índex de figures

2.1	Esquema GNU/Linux	6
2.2	Resum procés d'arrancada Linux	8
2.3	Estructura d'arxius de l'aplicació Simple-cdd	12
3.1	Diagrama cas d'ús i diagrama d'activitats de Xarxes de computadors 2	20
3.2	Diagrama cas d'ús i diagrama d'activitats de Transmissió de dades	21
4.1	Diagrama casos d'ús	27
4.2	Resum de les iteracions	29
4.3	Iteració base	30
4.4	Diagrama de flux del <i>script</i> de postinstal·lació	34
4.5	Iteracions de la darrera etapa	37
4.6	Moduls generador distribucions	38
4.7	Diagrama de flux del generador de noves versions de dEICbian	39
4.8	Diàleg de benvinguda del generador de distribucions	42
4.9	Diàleg de selecció de la llista de paquets a instal·lar	43
4.10	Diàleg final del generador	43
5.1	Planificació temporal	47

Índex de taules

2.1	Diferents tipus d'arxius en un sistema FHS.	10
2.2	Els directoris en un sistema FHS.	11
5.1	Costos	49
6.1	Resum de respostes	54

Capítol 1

Introducció

1.1 Motivacions

Les assignatures associades al departament d'Enginyeria de la informació i de les comunicacions tenen una sèrie de sessions pràctiques que es desenvolupen de manera completa o parcial als laboratoris d'aquest. Aquestes sessions pràctiques requereixen no només un programari determinat, sinó que aquest també estigui configurat de manera apropiada com el que es troba als laboratoris.

La metodologia actual de les sessions de pràctiques comporta que, en les sessions que no es desenvolupen de manera completa al laboratori, els alumnes han d'instal·lar i configurar programari en els seus ordinadors personals o bé connectar-se de forma remota als computadors del laboratori. Cal considerar que aquest últim supòsit produeix dificultats com ara treballar sense interfície gràfica, la compartició de recursos o que es produeixin desconexions per inactivitat. També s'ha de tenir en compte que en el cas que sigui necessari un canvi de programari aquest s'ha d'incloure i configurar a totes les màquines del laboratori i, en el cas que una d'aquestes s'espatlli s'ha de reinstal·lar de nou tot el sistema.

Actualment les sessions de laboratori del departament es realitzen en un entorn GNU/Linux, un sistema operatiu lliure amb una gran varietat de distribucions. La creació de distribucions GNU/Linux és habitual i té per objectiu oferir sistemes que s'adaptin a un conjunt d'usuaris determinat mitjançant la inclusió de programari i eines específiques. Les distribucions poden ser desenvolupades comercialment per empreses com ara Red Hat, Suse, Mandriva o Ubuntu. D'altra banda existeixen

distribucions com Debian o Gentoo basades exclusivament en programari lliure i desenvolupades per grups d'usuaris. Entre tot aquest conjunt de distribucions n'hi ha algunes que s'han pres com a base per realitzar altres distribucions i, gràcies a aquest fet, existeixen una sèrie d'eines pensades per realitzar noves distribucions personalitzades.

1.2 Objectius

Mitjançant l'estudi de la situació actual sobre la metodologia de les sessions pràctiques de les assignatures del departament i la tecnologia utilitzada i a l'abast, s'arriba a la conclusió que mitjançant una distribució GNU/Linux personalitzada es poden satisfer les necessitats del departament i solucionar en gran mesura els problemes actuals.

En primer lloc, es disposaria d'una distribució configurada especialment pels laboratoris amb el programari llest per ser utilitzat, el que facilitaria la reinstal·lació de les màquines.

En segon lloc, els alumnes podrien disposar d'una distribució portable que facilités la realització de les pràctiques en els seus ordinadors personals.

D'altra banda, si les necessitats canvien i nou programari és requerit per les assignatures o la distribució queda antiquada amb el pas del temps, es té la possibilitat de construir noves versions de la distribució.

D'aquesta manera, el projecte queda desglossat en els següents objectius concrets:

- Estudiar la personalització de distribucions.
- Estudiar les eines de desenvolupament de distribucions.
- Trobar les necessitats dels estudiants i membres del departament.
- Dissenyar un sistema que satisfaci les necessitats del departament.
- Implementar el sistema i realitzar les fases de test oportunes.
- Validar el sistema resultant.

Finalment, cal comentar que donat l'ús que tindrà la distribució no ha estat objectiu entrar en detalls d'implementació interna d'aquesta, sinó prendre de base

una distribució robusta i modificar-la de tal manera que compleixi tots els requisits necessaris del departament. En aquesta memòria s'hi podran trobar les diferents decisions que s'han pres respecte el disseny, les aplicacions utilitzades així com explicacions teòriques dels processos realitzats per aquestes.

1.3 Contingut de la memòria

La resta de memòria s'estructura mitjançant capítols que centraran l'atenció en aspectes concrets del projecte.

En el segon capítol s'introdueix teòricament el projecte i es veuran els conceptes GNU i Linux, la seva diferència i el paper que juguen en les distribucions GNU/Linux. També es veurà en aquest capítol les eines per al desenvolupament utilitzades durant la implementació del projecte.

El tercer capítol es centra en l'anàlisi de requisits del projecte, quedant definits els usuaris del sistema, la seva interacció amb aquest, els requisits funcionals associats a les assignatures i els no funcionals classificats per la seva tipologia. Finalment, es troba l'estudi de viabilitat del projecte.

En el quart capítol es troben totes les etapes del desenvolupament del projecte. En primer lloc s'explicarà la metodologia de desenvolupament que s'ha seguit: Un procés iteratiu. Acte seguit es detallaran les iteracions realitzades, començant pel disseny, passant per la implementació i acabant amb la fase de test del sistema.

El cinquè capítol tracta la planificació i costos del sistema on, mitjançant la comparació entre la planificació inicial i la real, s'analitzaran detalls concrets del desenvolupament.

En el sisè capítol es troben les conclusions on es resumeixen el producte resultant, els objectius que satisfà així com la validació dels resultats obtinguts, mitjançant un anàlisi estadístic d'enquestes realitzades a un conjunt d'usuaris. Finalment, es comenten possible línies de continuïtat del projecte.

Capítol 2

GNU, Linux i eines de desenvolupament

En aquest capítol es presenten continguts teòrics sobre GNU, Linux i les eines per al desenvolupament obtinguts durant la realització del projecte. Es farà especial èmfasi en la funció que realitzen tant els components de l'operatiu com les eines utilitzades, sense entrar en excessiu detall en el funcionament intern de cadascuna d'elles. Es prioritzarà, per tant, l'explicació de la funcionalitat davant de la implementació a baix nivell perquè és la primera la que cal per tal de realitzar, en primer lloc, l'anàlisi de requisits i, a continuació, cadascuna de les fases posteriors del desenvolupament.

2.1 GNU/Linux

Les distribucions GNU/Linux es basen en dos components fonamentals diferents: GNU i Linux. A continuació es descriuen totes dues.

2.1.1 GNU

El projecte GNU va néixer durant l'any 1984 amb l'objectiu de crear un sistema operatiu completament lliure que fos similar a UNIX. Aquest sistema es coneix amb el nom GNU [1]. Donat que el sistema GNU neix amb l'objectiu de ser similar i compatible amb UNIX, GNU disposa d'una serie de llibreries, aplicacions i eines de desenvolupament associades a un nucli independent, encarregat de la

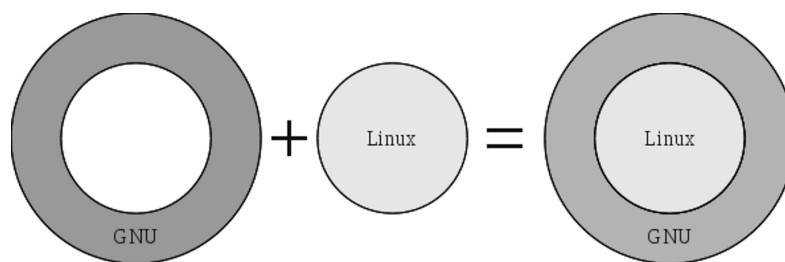


Figura 2.1: Esquema GNU/Linux

comunicació amb el hardware. Gran part d'aquestes llibreries, aplicacions i eines de desenvolupament ja estaven disponibles a UNIX i, per tant, es van adaptar. El sistema de finestres X Window o l'editor de textos TeX en són exemples. Com GNU, a diferència d'UNIX, va néixer sota la idea del software lliure es va alliberar el projecte sota llicència GPL que permetria a tots els usuaris executar, copiar, modificar i distribuir el sistema. Aquest darrer fet va augmentar significativament la seva popularitat.

L'any següent al seu naixement, el seu precursor, Richard Stallman, va fundar la Free Software Foundation - FSF - que oferiria suport legal i econòmic al projecte GNU. La popularitat del projecte va continuar augmentant ràpidament i diferents empreses varen començar a desenvolupar productes per GNU. Així a l'any 1990 l'operatiu comptava ja amb un editor de text propi, un compilador C i gran part de les utilitats del sistema operatiu UNIX. Encara mancava, però, el nucli del sistema. Per solucionar aquesta mancança es va inicialitzar el desenvolupament de Hurd, encara en desenvolupament, que havia de ser el nucli del sistema GNU. Tanmateix, i mentre Hurd no arribés, es va començar a utilitzar un nucli compatible amb el sistema GNU anomenat Linux. (Figura 2.1)

2.1.2 Linux

Linux va néixer l'any 1991 com a resultat d'un projecte personal de Linus Torvalds [13]. Aquest va créixer ràpidament i va convertir-se, gràcies a la col·laboració de diversos desenvolupadors [11] [12], en un nucli funcional amb milions de línies de codi [14]. Una de les decisions més importants va ser l'adopció de la llicència GPL per protegir aquest de l'explotació comercial i, d'aquesta manera, va sumar

el suport dels membres de la comunitat GNU.

El nucli Linux es pot definir com el cor del sistema operatiu GNU/Linux i s'encarrega de les funcions més importants d'aquest. Linux s'encarrega de gestionar la memòria de totes les aplicacions i processos en execució de l'operatiu, de l'administració del temps de processador que s'assigna als programes i processos en execució i, també, és l'encarregat de facilitar la comunicació entre l'usuari i els perifèrics de l'ordinador [2].

Arquitectura

Linux es considera un nucli monolític. En un nucli monolític tots els serveis bàsics es troben combinats al nucli. Tot i així a diferència dels nuclis monolítics tradicionals els controladors de dispositius i les seves extensions poden ser descarregats com a mòduls amb el sistema en funcionament. En canvi en l'altra variant arquitectònica, el *microkernel*, el nucli proporciona alguns serveis bàsics com ara la comunicació, els serveis d'entrada i sortida, la memòria o la gestió de processos però el serveis més específics es troben a la capa del *microkernel* [3].

Durant els inicis de Linux la seva arquitectura va ser molt discutida i encara és fruit de debat quin tipus d'arquitectura es més convenient [3]. De tota manera Linux es un nucli eficient en quant a gestió de memòria i processador així com d'una gran estabilitat. Precissament la seva eficiència i estabilitat han estat els punts forts que han fet que els sistemes operatius GNU/Linux siguin els més utilitzats entre els supercomputadors [10].

Un altre aspecte interessant del nucli es la seva portabilitat. Linux pot ser compilat per córrer en un gran nombre de processadors i plataformes diferents, cadascuna amb les seves restriccions i necessitats arquitectòniques.

En resum el nucli Linux és estable, portable i eficient.

2.1.3 Procés d'arrancada

Els sistema de procés d'arrancada a Linux es la manera en que els sistemes operatius basats en UNIX s'inicialitzen. El procés d'arrancada es porta a terme en diferents etapes i es poden classificar segons qui les realitza [15]. La primera etapa és la BIOS la que té el control del processador, després és el carregador d'arrancada qui en pren el control, en el següent pas és el nucli mateix qui en pren el control

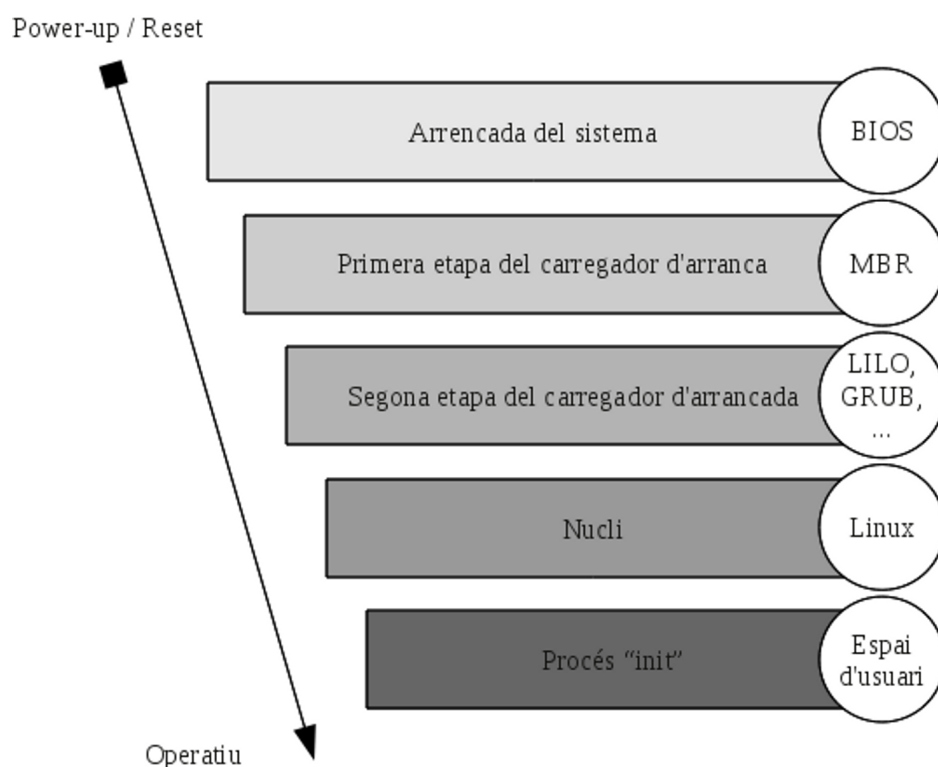


Figura 2.2: Resum procés d'arrancada Linux

i finalment són les aplicacions d'usuari, convivint amb el sistema operatiu, els que tenen control del processador. A la figura 2.2 es pot veure un resum d'aquest procés d'arrancada.

Durant la primera etapa, quan la BIOS té el control del processador, es realitzen operacions bàsiques de hardware. Aquest procés serà diferent segons l'arquitectura del processador i la BIOS.

Una vegada s'han realitzat les tasques de reconeixement de hardware, la BIOS carrega a memòria l'anomenat carregador d'arrancada, o *bootloader* en anglès, que pren el control de la inicialització de l'operatiu. La funció del carregador d'arrancada és la de carregar el sistema operatiu a memòria. Com el carregador d'arrancada no té suficient memòria com per carregar el sistema operatiu en una fase aquest consta de diverses etapes. La primera d'aquestes etapes és, en realitat,

la de carregar el mateix *bootloader* a memòria completament.

Hi ha diversos carregadors d'arrancada tot i que els més utilitzats en sistemes GNU/Linux són GRUB, LILO i el conjunt de carregadors SysLinux.

GRUB consta de dues etapes. La primera d'elles, ja descrita, s'utilitza per carregar la resta del carregador a memòria. La segona etapa executa el carregador i mostra el menú d'inici on es permet escollir el sistema en que es vol arrancar. Aquest es carregarà tot seguit a memòria i prendrà el control del procés d'arrancada [4].

LILO es un sistema més antic i segueix un procés molt similar a GRUB amb la diferència que els paràmetres de càrrega han de ser escrits a la memòria MBR al igual que tota la seva configuració [4].

El conjunt de carregadors d'arrancada SysLinux no s'utilitza habitualment per arrancar instal·lacions GNU/Linux completes perquè està destinat a sistemes d'arxius FAT, on normalment no s'instal·len distribucions GNU/Linux, però, en canvi, s'utilitza de manera freqüent en distribucions Live en sistemes de rescat [4].

Per tal de carregar el nucli a memòria, s'inicialitza un procés que es desenvolupa en dues etapes. La primera etapa s'encarrega de carregar el nucli i la segona fase s'encarrega de la seva execució. El nucli, que es troba emmagatzemat en un arxiu comprés a disc, es descomprimeix i es carrega a memòria. Mitjançant un disc RAM, un sistema de fitxers temporal utilitzat en la fase d'inicialització del nucli, es carreguen tots els drivers i les configuracions necessàries perquè el nucli prengui el control del processador. Una vegada el nucli es troba totalment operatiu, aquest pren part de la seva inicialització. El nucli, un cop s'ha carregat completament a memòria, inicialitza un procés que s'encarregarà d'assignar l'espai d'usuari, tots els processos necessaris per l'entorn d'usuari i el sistema d'autenticació d'usuaris. Aquest procés anomenat procés 'init' rep un paràmetre anomenat runlevel que defineix l'entorn de carrega on s'hi troben *scripts* necessaris per la inicialització de diferents aplicacions. Quan aquest procés finalitza ho fa també el procés de carrega de l'operatiu.

2.1.4 El sistema d'arxius

El sistema d'arxius de Linux es basa en l'estàndard FHS, tot i que la gran majoria de distribucions no el compleixin amb plenitud. L'estàndard FHS defineix una

	Compartit	Restringit
Estàtic	/usr	/etc
	/opt	/boot
Variable	/var/mail	/var/run
	/var/spool/news	/var/lock

Taula 2.1: Diferents tipus d'arxius en un sistema FHS.

normativa on s'estableix els directoris principals i el seu contingut en sistemes de la família UNIX. Aquesta definició del sistema té com a base l'estàndard FSSTND [5] i es va desenvolupar amb l'idea de crear una versió més comprensiva i adaptada a tots els sistemes UNIX que aquest.

En els sistemes UNIX tots els directoris i arxius apareixen, tot i que es trobin en dispositius diferents, sota el node arrel “/” que es el contenidor de tot el sistema de jerarquia.

Tal com es defineix al segon capítol del document d'especificació de l'estàndard existeixen quatre grans tipus de arxius als sistemes UNIX. Compartits o restringits i variables o estàtics. [6]

Els arxius compartits son aquells que poden ser emmagatzemats en un *host* i utilitzats en un altre. Els arxius restringits son aquells que no són compartits. Per exemple els arxius que es troben al “/home” són compartits mentre que els dels dispositius bloquejats no ho són. Els arxius estàtics inclouen binaris, llibreries, documentació i altres arxius que no poden ser modificats sense la intervenció de l'administrador. Els arxius variables són aquells que no són estàtics.

A la taula 2.1 extreta del document d'especificacions de l'estàndard es pot trobar un exemple dels diferents tipus d'arxius en un sistema que compleix l'estàndard.

Com s'ha mencionat anteriorment a l'estàndard també queden definits quins són els directoris que formen part del node arrel. A la taula 2.2 s'hi pot trobar quins són aquests directoris i un petit resum sobre la seva funció al sistema.

Directori	Descripció
bin	Comandes binàries essencials
boot	Arxius estàtics del carregador d'arrancada
dev	Arxius de dispositius
etc	Configuració específica del sistema amfitrió
lib	Llibreries essencials compartides i mòduls del nucli
media	Punt de muntatge per dispositius extraïbles
mnt	Punt de muntatge temporal per sistemes de fitxers
opt	Paquets de software d'aplicacions addicionals
sbin	Comandes binàries essencials de sistema
srv	Dades per serveis proporcionats pel sistema
tmp	Arxius temporals
usr	Jerarquia secundària
var	Dades variables

Taula 2.2: Els directoris en un sistema FHS.

2.2 Eines per al desenvolupament

En aquesta secció s'explica el funcionament de les diferents aplicacions utilitzades durant el desenvolupament de la distribució. Aquestes eines de desenvolupament son utilitzables en les distribucions Debian, un dels requisits no funcionals del projecte és el motiu pel qual la distribució s'ha realitzat amb aquest sistema base tal com s'explicarà en el següent capítol de la memòria. Tot i això cal destacar que Debian es un sistema robust amb el que es basen gran nombre de distribucions, entre elles una de les més utilitzades: Ubuntu, i que reuneix tot un seguit d'eines per al desenvolupament que la fan apropiada com a base per la creació de distribucions.

2.2.1 Simple-cdd (Custom Debian Distribution)

L'aplicació Simple-cdd permet la creació d'un instal·lador de Debian personalitzat afegint a aquest paquets i *scripts* de configuració i instal·lació propis [7]. En primer lloc i abans d'entrar en detall en el seu funcionament intern cal explicar com s'executa i quina configuració requereix.

En primer lloc cal generar un arxiu de configuració del programa que segueix una sintaxi específica que es detallarà amb cura al quart capítol de la memòria. Aquest arxiu de configuració el rebrà per paràmetre l'aplicació i entre d'altres coses

▼ deicbian_instalador	3 elements	carpeta
▼ images	1 elemento	carpeta
debian-504-i386-CD-1.iso	575,8 MiB	imagen de CD en bruto
▼ profiles	3 elements	carpeta
deicbian.packages	240 bytes	documento de texto sencillo
deicbian.postinst	7,5 KiB	documento de texto sencillo
deicbian.preseed	7,5 KiB	documento de texto sencillo
simple-cdd.conf	1,2 KiB	documento de texto sencillo

Figura 2.3: Estructura d'arxius de l'aplicació Simple-cdd

servirà per establir el nom de la distribució, l'arquitectura d'aquesta o els miralls que es faran servir durant l'instal·lació.

L'aplicació també farà ús d'una carpeta anomenada “profiles” on es trobaran la llista de paquets a instal·lar, els *scripts* de post-instal·lació i els arxius de *preseed*, que es fan servir per automatitzar algunes fases del procés d'instal·lació.

L'aplicació genera finalment una carpeta temporal “tmp” on queden inclosos tots els arxius necessaris per generar la imatge ISO que es podrà trobar a la carpeta “images”.

L'estructura de la carpeta des de on es fa la crida a l'aplicació ha de tenir un aspecte semblant a la figura 2.3.

Un cop s'han donat a conèixer tant l'estructura com els arxius necessaris per l'aplicació ens centrarem en el funcionament d'aquesta detallant cadascun dels passos que fa.

Quan es crida l'aplicació el primer que es fa és una etapa anomenada *bootstrap* on es genera a una carpeta temporal els mínims arxius necessaris d'un instal·lador Debian.

Tot seguit s'afegeixen a aquesta carpeta temporal tots els paquets, i les seves dependències, dels perfils que es troben a la carpeta “profiles”. Aquest procés consisteix en realitat en l'elaboració d'un mirall parcial de Debian amb els paquets necessaris per l'instal·lació del sistema.

Acabat aquest procés d'estudi de dependències i recepció de paquets s'afegeixen els arxius locals indicats a la configuració a la carpeta temporal.

Un cop a la carpeta temporal es troba l'estructura exacta d'un instal·lador Debian es comença el procés de generació de la imatge ISO.

Tots els passos realitzats per l'aplicació es poden realitzar mitjançant un *script* propi i diferents aplicacions de baix nivell incloses als paquets de desenvolupadors de Debian però, donat que l'aplicació existeix i no afegeix cap inconvenient en la generació de l'instal·lador, s'ha considerat innecessari realitzar una implementació pròpia.

2.2.2 Live Helper

L'aplicació Live-helper està destinada a l'elaboració de distribucions *live* Debian de forma senzilla però potent gràcies a la possibilitat de utilitzar-la amb diferents nivells d'abstracció [8]. En el desenvolupament del projecte aquesta s'ha utilitzat en el seu nivell més baix per tal de poder aprofitar les avantatges en quant a personalització que això permet. A continuació s'expliquen les diferents crides a l'aplicació analitzant que fa cadascuna deixant per el capítol quatre de la memòria l'explicació de quin ha estat el seu ús en el projecte.

lh config: Mitjançant aquesta comanda i una serie de paràmetres es generen els arxius de configuració necessaris per la utilització de l'aplicació. Per exemple es poden configurar el gestor de paquets a utilitzar ("apt" o "aptitude"), quins miralls i repositoris, o fonts de programari, es faran servir i fins i tot indicar uns perfils bàsics que permetrien la generació del *live* de manera completament automatitzada.

lh build: La comanda "lh build" genera de manera automatitzada la distribució excepte en el cas que s'apliqui el paràmetre "- interactive" que permet mitjançant una pausa afegir i personalitzar el *live*. Mitjançant l'aplicació de "lh config" i "lh build" es pot generar un *live* DVD funcional de Debian. Aquesta comanda es pot obviar durant el procés de creació de la distribució utilitzant instruccions de més baix nivell com les que es detallen a continuació.

lh bootstrap: Genera l'estructura de fitxers necessària d'un sistema Debian dins de la carpeta "chroot" a l'entorn de treball. Aquesta etapa és similar a l'etapa de *bootstrap* de l'aplicació Simple-cdd. La diferència bàsica consisteix en que l'etapa de *bootstrap* de l'aplicació Simple-Cdd construeix el sistema de fitxers necessari per Debian-installer, en canvi, l'etapa de *bootstrap* de l'aplicació Live-Helper construeix el sistema de fitxers d'un sistema Debian instal·lat, que es el

que requereix un sistema *live*. Aquest sistema de fitxers generat és exactament el comentat anteriorment i que es troba resumit a la taula 2.2.

lh chroot: Mitjançant “chroot” s’emula un sistema Debian instal·lat, el generat per la comanda “lh bootstrap”, perquè s’hi puguin fer les modificacions necessàries i generar d’aquesta manera la distribució personalitzada. Això requereix una sèrie de configuracions com ara configurar el gestor de repositoris o els dispositius i que la comanda realitza de manera automàtica mitjançant una sèrie de paràmetres.

lh binary: Mitjançant “squashfs” comprimeix l’estructura de fitxers de la carpeta “chroot”. L’arxiu comprés serà el sistema de fitxers del *live* DVD. Un cop acabat aquest procés de compressió es genera la imatge ISO de la distribució.

lh binary_iso: Genera únicament la imatge ISO de la distribució sense realitzar la compressió del sistema de fitxers. Aquesta comanda s’utilitza per generar la imatge ISO si ja s’ha comprés el sistema de fitxers i no s’han realitzat canvis en aquest.

lh binary_usb: Genera únicament la imatge IMG de la distribució perquè aquesta sigui utilitzada per arrancar des de una memòria USB.

Existeixen moltes altres comandes per realitzar diferents funcions que no s’han trobat d’utilitat pel desenvolupament del projecte o que, donada la seva simplicitat, els processos automatitzats per aquestes s’han realitzat de manera manual.

En resum, i tal com suggereix el nom de l’aplicació, Live-Helper consisteix en un ventall de comandes que faciliten la creació d’un sistema *live* personalitzat basat en Debian.

2.2.3 Zenity

Zenity es una eina que permet la realització d’interfícies gràfiques associades a *scripts* en *bash* desenvolupada per entorns d’escriptori Gnome mitjançant la llibreria de GTK+.

Mitjançant Zenity es poden crear el següents tipus de diàlegs: [9]

Calendaris: Permeten escollir una data.

Textos d’entrada: Mitjançant un diàleg d’entrada s’associa una cadena de caràcters com a paràmetre del *script*.

Missatges d’error: Permeten donar un avís d’error.

Selecció d’arxius: Mitjançant un diàleg d’entrada s’associa a un paràmetre del

script una ruta del sistema.

Llistes: Permeten crear un diàleg en forma de llista.

Notificacions: Generació d'una icona a l'àrea de notificació del sistema.

Barres de progrés: Creació d'una barra de progrés associada al *script*.

Diàlegs YES/NO: Es poden fer preguntes al usuari que tenen per resposta un booleà.

Alertes: Permeten donar un avís i continuar l'execució del *script*.

Les crides a les diferents funcionalitats de Zenity es fa dins del propi *script* que rep les respostes associades als diàlegs en forma de paràmetres.

2.2.4 Màquines virtuals

Les màquines virtuals són aplicacions que emulen un computador i que poden executar processos com si fossin un de real. Hi ha múltiples aplicacions i tipus de màquines virtuals i els seus usos varien segons la seva tipologia però, en el desenvolupament d'aquest projecte, han estat màquines virtuals de sistema les que s'han utilitzat amb l'objectiu de provar la distribució generada. Les màquines virtuals de sistema permeten multiplexar-se i utilitzar en cadascuna d'elles un sistema operatiu diferent.

Existeixen diverses aplicacions que permeten la creació de màquines virtuals però en el desenvolupament del projecte s'han vist bàsicament tres: VirtualBox OSE, VmWare Player i Qemu.

La primera consisteix en la versió de codi lliure de VirtualBox [16]. Una aplicació per gestionar màquines virtuals fàcil d'utilitzar amb una interfície gràfica potent que permet configurar diversos aspectes de la màquina virtual com la memòria destinada a aquesta, opcions gràfiques o l'entrada d'imatges ISO com a CD/DVD.

La segona aplicació forma part de l'aplicació privativa VmWare [17] i aquesta versió distribuïda de manera gratuïta no permet la creació de màquines virtuals, només la seva execució.

Finalment la tercera aplicació, Qemu [18], permet la creació i execució de màquines virtuals. No té interfície gràfica i la seva configuració es realitza mitjançant la terminal.

Com es comentava anteriorment en el desenvolupament del procés han estat

especialment útils per realitzar les fases de proves de les diferents iteracions de la distribució com s'explicarà al capítol quatre de la memòria.

Capítol 3

Anàlisi de requisits

En aquest capítol es detallen els requisits que es van establir per la distribució d'EICbian mitjançant la col·laboració del tècnic de suport del departament així com diferents professors de pràctiques d'assignatures impartides als laboratoris de docència del departament. L'anàlisi de requisits s'ha desenvolupat seguint la metodologia IEEE ANSI 830-1993, tot i que adaptada a les necessitats d'aquest projecte. Abans d'entrar en detall cal comentar que el principal requisit, establert pel tècnic del departament, ha estat realitzar la distribució basada en Debian. Una de les decisions importants del disseny, per tant, quedava tancada d'inici. La resta de requisits en general estan submisos al programari utilitzat en les sessions de pràctiques de les assignatures impartides pel departament.

3.1 Introducció

Aquestes especificacions tenen com objectiu analitzar i documentar les necessitats del sistema. Per aquest motiu, s'identificaran els requisits a satisfer mitjançant entrevistes via correu electrònic, l'estudi dels problemes a solucionar així com establir prioritats de cara a validar el sistema final desenvolupat. L'objectiu és tenir documentació suficient per comprovar que el sistema desenvolupat compleixi totes les necessitats del usuari.

3.2 Identificació d'usuaris i participants

L'objectiu d'aquesta secció és la d'identificar els usuaris implicats en el sistema. Aquests són:

- Alumnes: Format pels alumnes matriculats a les assignatures que tenen sessions de pràctiques als laboratoris de docència del departament d'enginyeria de la informació i de les comunicacions.
- Professors: Format pels docents de les assignatures del departament que tenen sessions de pràctiques als laboratoris de docència d'aquest.
- Tècnics de suport del departament: Format pels tècnics de suport del departament encarregats, entre d'altres coses, del manteniment dels laboratoris de docència del departament.

Finalment, cal tenir en compte que els usuaris del sistema seràn alumnes o professors d'Enginyeria informàtica i Enginyeria de telecomunicacions i, per tant, tindran coneixements suficients per utilitzar de manera apropiada la distribució. De tota manera, seria apropiat realitzar una documentació de la distribució per detallar aspectes específics d'aquesta.

3.3 Catàleg de requisits del sistema

L'objectiu de l'especificació del sistema es definir de manera clara, precisa i completa totes i cadascuna de les funcionalitats i restriccions del sistema.

3.3.1 Objectius i abast del sistema

Els principal objectiu consisteix en desenvolupar una distribució GNU/Linux basada en Debian que inclogui el programari necessari, instal·lat i configurat de manera adient, per realitzar totes les sessions de pràctiques que s'imparteixen als laboratoris de docència del departament. Aquesta distribució portarà el nom de dEICbian.

Els laboratoris de docència del departament compten amb 26 ordinadors amb els següents processadors: "Intel(R) Core(TM)2 Duo CPU E7400 @ 2.80GHz" i "Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz". En els laboratoris es desenvolupen pràctiques de 7 assignatures diferents cadascuna d'elles amb diferents ne-

cessitats. Aquestes necessitats es consideraran els requisits funcionals del sistema. També s'ha de tenir en compte que algunes assignatures es poden desenvolupar parcialment fora dels laboratoris i, per tant, seria convenient que la distribució pogués inicialitzar-se o instal·lar-se també fora d'aquests. Igualment cal tenir en compte que els ordinadors del laboratori poden canviar en el futur. Per tant el sistema ha de ser portable.

Els sistema ha de ser també escalable. Els requisits de les assignatures variaran segur amb el pas del temps i per tant s'ha de poder fer front a aquesta necessitat.

3.3.2 Descripció general del sistema

El sistema consisteix en una distribució GNU/Linux anomenada dEICbian que aglutina tot el programari necessari de les assignatures. També ha de formar part del sistema una aplicació que permeti actualitzar la distribució dEICbian per fer fronts a noves necessitats que puguin sorgir en el futur.

3.3.3 Requisits funcionals

Per tal de detallar els requisits funcionals del sistema, abans, cal llistar les assignatures que imparteixen sessions pràctiques als laboratoris del departament donat que el requisits funcionals estaran condicionats per cadascuna de les assignatures. Per tal de estudiar els requisits funcionals de les assignatures s'han elaborat diagrames d'ús de la distribució com els que es poden veure, a mode d'exemple, a les figures 3.1 i 3.2.

Les assignatures en que les sessions de pràctiques es donen als laboratoris de docència del departament són:

- **Xarxes de computadors I i II:** Per la realització de les pràctiques de Xarxes de computadors és necessari incloure un compilador de C a la distribució, un editor de texts avançat, destinat a la programació, i per facilitar a l'alumne la compilació i edició del codi font de les pràctiques una IDE de C. També es necessari un entorn de programació en Java i, a més a més, donat que els enunciats de l'assignatura es troben en format pdf serà necessari un lector que permeti obrir aquest format. També s'ha de tenir en compte la generació de previs i informes de pràctiques pels que serà necessari una suite ofimàtica.

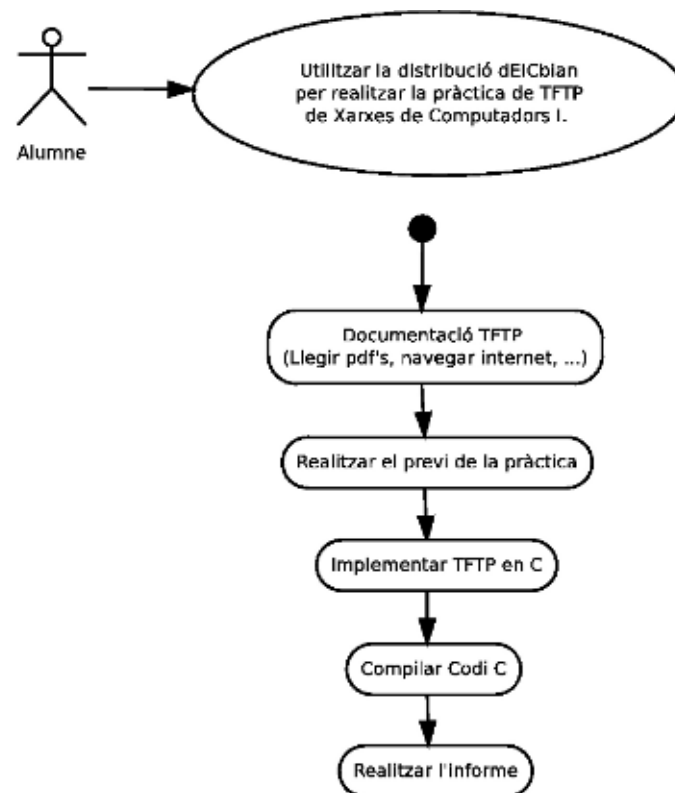


Figura 3.1: Diagrama cas d'ús i diagrama d'activitats de Xarxes de computadors 2

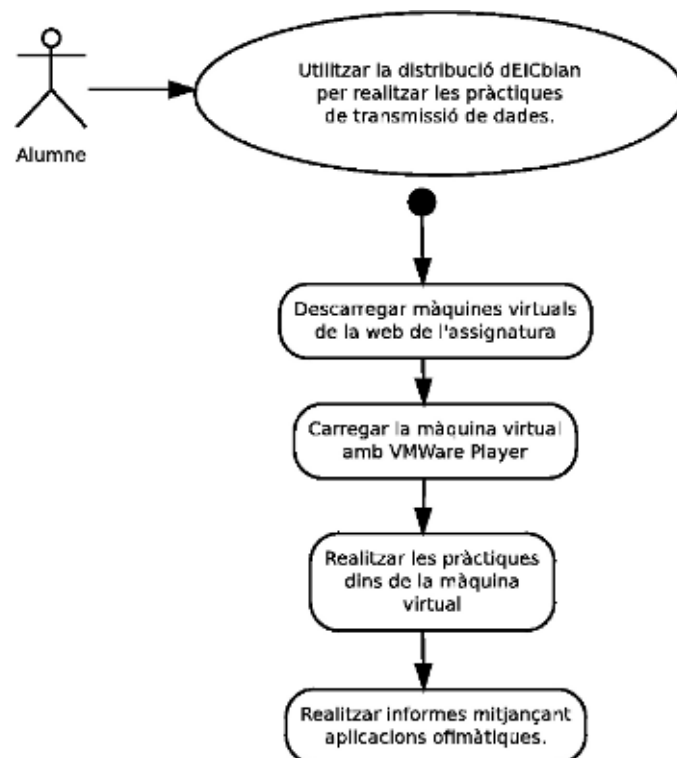


Figura 3.2: Diagrama cas d'ús i diagrama d'activitats de Transmissió de dades

D'altra banda, també serà necessari un analitzador de protocols anomenat Wireshark.

- **Xarxes de computadors (telecomunicacions):** Les pràctiques de Xarxes de computadors de enginyeria de telecomunicacions requereixen algunes eines útils de les distribucions GNU/Linux com ara lsof/fuser, strace, ltrace i gdb.
- **Transmissió de dades:** Les pràctiques de transmissió de dades es realitzen sobre màquines virtuals de VmWare per aquest motiu és necessari un reproductor de màquines virtuals de VmWare.
- **Teoria de la codificació:** Les pràctiques de Teoria de la codificació es realitzen íntegrament en magma.
- **Seguretat Computacional:** Les pràctiques de Seguretat Computacional no més requereixen un entorn de programació en Java i eines bàsiques de les distribucions GNU/Linux.
- **Criptografia i seguretat en xarxes:** Les pràctiques de Criptografia i seguretat en xarxes requereixen el següent programari: nessus, sshd, iptables i snort.

3.3.4 Requisits no funcionals

Requisits d'interfícies externes

Les interfícies de la distribució dEICbian són les mateixes que les d'un sistema Debian i no s'hi troba cap requisit excepte la de la inclusió d'una GUI gràfica que permeti la interacció amb les aplicacions de manera còmoda.

La interfície externa del generador de distribucions ha d'estar pensada per ser executada en una distribució dEICbian i els seus dispositius d'entrada/sortida bàsics seran monitor, ratolí i teclat.

Requisits de disseny

Com es comentava a la introducció d'aquest capítol, ha estat un requisit de disseny la tria d'una distribució Debian com a base de la distribució personalitzada.

Requisits de rendiment i fiabilitat

Tot i no haver-hi uns requisits de rendiment gaire específics, òbviament, s'ha de tenir en compte el propòsit de la distribució i per tant s'hauran de prioritzar la fiabilitat i rendiment del sistema davant de efectes estètics i altres detalls que no millorin la usabilitat del sistema.

Requisits de portabilitat

La distribució ha de permetre la seva instal·lació fora del domini d'ordinadors de l'autònoma així com el seu ús de manera poc complexa i sense requerir temps elevats de configuració o instal·lació.

Requisits legals

La distribució estarà disponible per als alumnes de l'Escola d'Enginyeria i, per tant, s'han de tenir en compte els requisits legals que comporta distribuir programari. Per tant d'evitar il·legalitats el programari inclòs en la distribució ha de ser o bé programari lliure o bé programes de les que es disposin les llicències apropiades.

3.4 Resum

En el següent llistat es troben resumits els requisits fonamentals del projecte:

- La distribució d'EICbian ha d'estar basada en Debian.
- S'han de tenir en compte els requisits i necessitats de cadascuna de les assignatures impartides als laboratoris de docència del departament d'Enginyeria de la informació i de les comunicacions.
- El sistema ha de ser portable, és a dir, ha de poder ser arrancat en ordinadors amb una gran varietat de hardware diferent.
- Per tal de garantir l'escalabilitat del sistema per tal de donar opció a actualitzar el sistema a noves necessitats.
- Donada la naturalesa del projecte s'haurà de prioritzar l'ús de programari lliure i no es podrà incloure a la distribució cap programari privatiu del que no es disposin les llicències apropiades.

3.5 Estudi de la viabilitat del projecte

Per tal de realitzar l'estudi de viabilitat del projecte s'han analitzat diversos punts: La viabilitat tècnica, la viabilitat econòmica i la viabilitat legal.

En primer lloc, s'ha estudiat la viabilitat tècnica. Com s'ha comentat a l'introducció de la memòria existeixen diverses distribucions personalitzades basades en Debian, per tant, es pot afirmar que tècnicament es viable realitzar el projecte. S'ha de considerar també el fet que el programari necessari per realitzar les pràctiques és conegut, finit i no gaire extens, així, realitzar una distribució que l'inclogui és possible. Per acabar aquest estudi, afegir que no calen dispositius especials i els medis necessaris, com els equips de laboratori o la distribució base: Debian, existeixen i són disponibles.

En segon lloc, i pel que fa la viabilitat econòmica, al no requerir cap altre dispositiu que els equips de laboratori i algun ordinador personal per tal de realitzar les proves, el cost del projecte és baix i viable. Tot i que a simple vista es podria afirmar la viabilitat econòmica del projecte donat els recursos requerits, es va desenvolupar un pressupost orientatiu per mirar d'estudiar-ho. Aquest pressupost orientatiu juntament amb una comparació d'aquest amb el pressupost real del projecte es troben al capítol cinquè de la memòria.

Finalment, pel que fa l'apartat legal, remarcar que es disposen totes les llicències del programari que inclourà la distribució així com del programari que s'utilitzarà durant la creació d'aquesta. A més a més no es recolliran, no s'emmagatzemaran dades personals ni s'infringiran patents per tant el projecte es troba dins del marc de la legalitat espanyola.

Un cop analitzats cadascun dels punts corresponents a l'estudi de viabilitat, podem afirmar que el projecte és viable.

Capítol 4

Disseny i desenvolupament

En aquest capítol s'expliquen acuradament el desenvolupament de la distribució, les decisions de disseny preses i quin ha estat l'ús que se'ls hi ha donat a les eines per al desenvolupament esmentades al segon capítol de la memòria.

4.1 Metodologia de desenvolupament

En aquest apartat s'explicarà la metodologia de desenvolupament iteratiu [19] que s'ha seguit durant la realització del projecte. El cicle de vida d'aquest model de desenvolupament consisteix en una etapa inicial, o iteració base, on es compleixen els requisits bàsics del projecte i en etapes iteratives posteriors, on s'afegeixen funcionalitats al sistema. Tot i que donada la naturalesa del projecte no s'ha pogut comptar amb l'opinió dels usuaris finals del sistema, al finalitzar cada iteració, sí que s'han elaborat proves periòdicament amb diferents persones i ordinadors.

Per tal d'entendre aquest procés, es detallaran, en les seccions d'aquest capítol de la memòria, quin ha estat el disseny, la implementació i les proves de les diferents iteracions. Cal destacar que el disseny de la iteració no consisteix en un disseny del sistema, ja realitzat amb anterioritat, sinó a detallar quines funcionalitats s'afegiran al sistema durant la iteració. En aquest aspecte també cal dissenyar quin serà el sistema que s'implementarà durant la iteració inicial que formarà el sistema base sobre el que començar el desenvolupament.

A mode d'introducció a tot aquest procés, es comenta, de manera general, quin ha estat l'evolució del desenvolupament. En la etapa inicial es genera un

instal·lador de una distribució personalitzada Debian bàsica, és a dir, sense cap funcionalitat extra ni cap de menys. Aquest primer pas es considera especialment útil donat que es comença a conèixer d'aquesta manera les eines per al desenvolupament que es fan servir al llarg de tot el projecte. Un cop acabada aquesta primera etapa inicial i amb un producte funcional com a resultat es comença el procés iteratiu en que al llarg de diferents etapes es va afegint progressivament el programari necessari per satisfer tots els requisits, els elements personalitzats de la distribució i finalment, i també de manera progressiva, el live de la distribució, al que anomenarem "Live dEICbian". Finalment la creació de dEICbian que consisteix en la unió de l'instal·lador dEICbian i Live dEICbian en un únic DVD.

4.2 Disseny

En primer lloc, per tal de desenvolupar el disseny de la distribució, s'ha realitzat un diagrama de casos d'ús general on es detallen les interaccions amb els usuaris del sistema. Aquest diagrama de casos d'ús general es troba a la figura 4.1.

Els usuaris del sistema són els alumnes, professors i els tècnics de suport del departament. Els primers utilitzaran la distribució als laboratoris per realitzar les sessions de pràctiques i, en el cas que aquestes es puguin desenvolupar fora dels laboratoris, podran fer servir la distribució a casa per tal d'utilitzar el mateix sistema amb el que han treballat al laboratori.

Els professors tanmateix podran utilitzar la distribució durant les sessions de laboratori per avaluar i realitzar les tasques docents corresponents a les sessions. D'altra banda, els professors d'acord amb les necessitats que puguin sorgir en el futur tindran a la seva disposició un generador de noves versions de la distribució que permetrà afegir nous paquets de programari de cara a fer front a futures necessitats, com per exemple noves pràctiques. Aquesta darrera funció també és competència del tècnic de suport del departament.

Els tècnics de suport s'encarregaran de l'actualització periòdica del sistema mitjançant l'ús del generador de noves versions de la distribució. Mitjançant l'actualització es podran corregir mancances o errors i també servirà per mantenir un sistema modern i dinàmic com són les distribucions GNU/Linux.

Per tal de començar el disseny de la distribució, cal escollir el programari que formàra part d'aquesta mitjançant l'estudi dels requisits funcionals del sistema. En

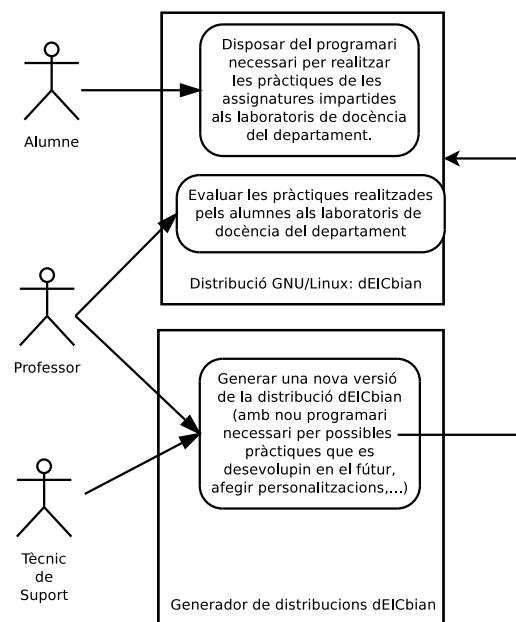


Figura 4.1: Diagrama casos d'ús

alguns casos els requisits funcionals detallen aplicacions concretes però en altres es permet escollir d'un ventall ampli de aplicacions que realitzen funcions similars.

Una de les decisions importants del disseny es escollir l'entorn d'escriptori perquè aquest serà el medi comunicatiu entre l'usuari i el sistema. L'únic requisit d'aquest és utilitzar un sistema de finestres senzill d'utilitzar i que inclogui interfície gràfica. S'opta per l'ús de XFCE4, un entorn d'escriptori lleuger, fàcil d'utilitzar i atractiu visualment. La seu principal virtut es el lleu consum de recursos i, alhora, la velocitat de càrrega. Els inconvenients d'aquest entorn d'escriptori no afecten a la seva elecció donat que no és necessària la potència d'entorns d'escriptori com KDE o Gnome.

Pel que fa el navegador web s'escull incloure a la distribució la versió lliure del navegador Firefox anomenat iceWeasel. S'escull aquest navegador, en primer lloc, perquè, tal com s'acaba de dir, forma part de la comunitat de programari lliure i, en segon lloc, per la popularitat del navegador Firefox, al que n'és pràcticament idèntic.

La suite ofimàtica escollida és OpenOffice donada la seva popularitat i qualitat. Només amb l'instal·lació d'aquesta suite es satisfan tots els requisits ofimàtics.

La resta de decisions preses sobre el programari són menors i, en general, s'han facilitat varies alternatives per donar a l'usuari variabilitat per escollir l'aplicació que li resulti més còmoda. D'aquesta manera s'han seleccionat diferents lectors de arxius PDF per donar opcions a l'usuari davant la impossibilitat d'obrir determinats formats per part del ventall de programari lliure destinat a aquesta funció. També s'han inclòs diferents gestors de compressió pel mateix motiu.

Pel que fa la resta de programari com ara els entorns de desenvolupament, o *IDE's*, s'ha escollit aquells que s'utilitzen actualment a les sessions de laboratori o aquells que els professors van suggerir o decidir durant el procés d'entrevistes via correu electrònic.

Només "magma", entre el programari necessari, no es pot incloure a la distribució per aspectes legals. Per tal de solucionar aquesta mancança es decideix, d'acord amb els professors de pràctiques de Teoria de la Codificació, assignatura que fa ús de l'aplicació, executar aquesta de manera remota.

La resta de aspectes de disseny, força més concrets, s'aniran comentant en les seccions posteriors de la memòria conforme apareguin en les diferents iteracions del desenvolupament.

4.3 La distribució: dEICbian

La distribució anomenada dEICbian s'ha generat de manera seqüencial afegint diferents parts fins a formar el mòdul complet seguint el mètode d'un procés iteratiu. El resultat ha estat un live DVD instal·lable de la distribució. El procés de generació de la versió instal·lable i Live dEICbian han estat independents tot i que tots dos formin part de dEICbian.

Per tal de comentar quin ha estat el procés de creació de la distribució dEICbian, s'explicarà per cada fase o iteració de desenvolupament les seves fases de disseny, implementació i test. Amb l'objectiu d'organitzar aquestes fases de desenvolupament, s'han dividit les iteracions en tres etapes diferents.

La primera etapa correspon des de la iteració base fins al desenvolupament de l'instal·lador de la distribució dEICbian. La segona etapa comença amb la finalització del instal·lador dEICbian i acaba amb el resultat de la generació de Live dEICbian. Finalment, la darrera etapa consta del procés d'afegir l'instal·lador de la distribució a Live dEICbian i, també, del desenvolupament del generador per tal

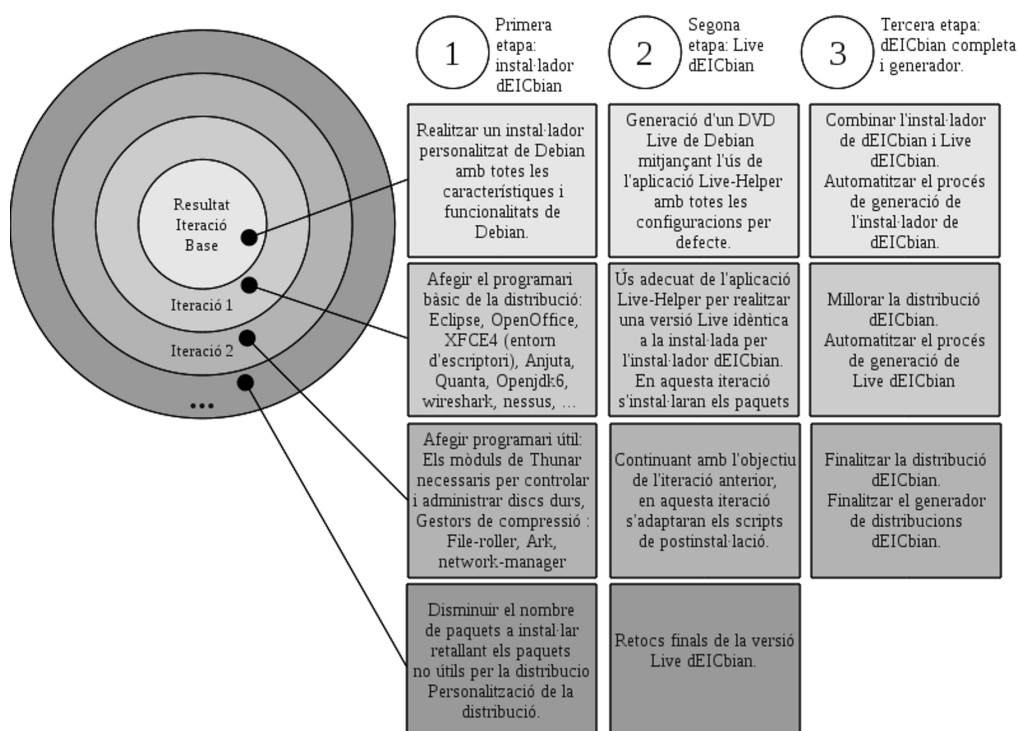


Figura 4.2: Resum de les iteracions

d'automatitzar el procés de generació de distribucions dEICbian.

A la figura 4.2 s'hi pot trobar un resum dels dissenys de les iteracions de les diferents etapes que es descriuran a continuació.

4.3.1 De l'iteració base a la generació de l'instal·lador dEICbian

En les següents subseccions es documentarà acuradament el procés de generació de la distribució. Aquestes seccions són la base de realització i, per tant, cal detallar àmpliament els passos realitzats per tal de garantir la documentació de cara a una possible continuïtat del projecte.

La iteració base

El disseny de la iteració base tenia per objectiu no només generar una primera distribució funcional sinó que també aprendre i conèixer les diferents eines per al desenvolupament que s'utilitzarien al llarg del projecte. D'aquesta manera es

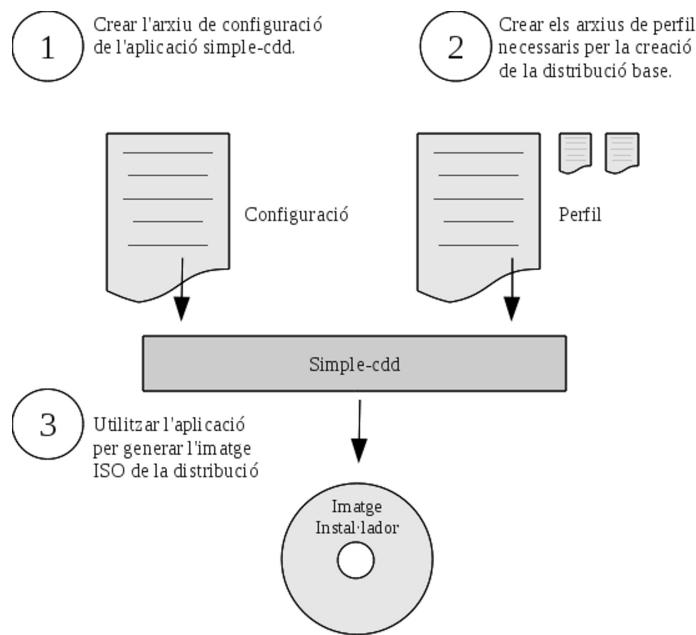


Figura 4.3: Iteració base

varen definir que els objectius del producte resultant de la iteració base seria obtenir un instal·lador d'una distribució personalitzada Debian amb totes les aplicacions i funcionalitats de Debian.

Durant el desenvolupament de la iteració base, cal utilitzar per primer cop l'aplicació Simple-cdd [7] de manera que, abans, s'ha de configurar de manera apropiada i també s'han de fer els primers perfils necessaris. Els passos a seguir per tal de desenvolupar la iteració base i, en definitiva, per utilitzar adequadament l'aplicació Simple-cdd es troben a la figura 4.3.1.

En primer lloc, cal explicar quin és el contingut de l'arxiu de configuració. L'arxiu de configuració conté opcions que l'aplicació utilitzarà per generar la distribució de manera apropiada. Una de les propietats que cal definir és quin perfil s'utilitzarà per generar la distribució. En els perfils s'hi troben els paquets de programari a instal·lar, els *scripts* de postinstal·lació i els anomenats *preseeds* que són utilitzats per generar respostes automàtiques a preguntes del instal·lador. Per definir quins perfils formaran part de la distribució cal afegir una línia a l'arxiu de configuració amb la següent sintaxis:

```
profiles="nom_perfil_1 nom_perfil_2 ... nom_perfil_n"
```

Els perfils afegits d'aquesta manera formaran part del instal·lador de la distribució però no s'instal·laran de manera automàtica, si es vol que així sigui, caldrà afegir els perfils a l'arxiu de configuració amb la següent sintaxis:

```
auto_profiles="nom_perf_1 nom_perf_2 ... nom_perf_n"
```

El perfil “default”, corresponent a la configuració de perfil necessari per realitzar la distribució base, s'inclou a l'arxiu de configuració base amb totes dues propietats, es a dir, formarà part de la distribució i s'instal·larà de manera automàtica.

També es poden definir l'idioma i el país de l'instal·lador mitjançant l'assignació d'una cadena de caràcters amb l'idioma i el país escollit a l'etiqueta “language” i “country” respectivament.

Aquestes propietats no es varen considerar necessàries alhora de desenvolupar la iteració base.

D'igual manera s'hi poden definir quins seran els miralls utilitzats per descarregar els paquets que formaran part de la distribució mitjançant l'assignació de les etiquetes “debian_mirror” i “security_mirror”.

La configuració dels miralls es necessària però, tret de problemes puntuals dels servidors que els allotgen, no hi ha cap diferencia, a excepció de la velocitat de descarrega de paquets, entre les diferents opcions. Els miralls dels repositoris de paquets Debian es poden trobar a la pàgina web oficial de la distribució [20].

L'última opció utilitzada en el procés de generació de la distribució base és la de definir l'arquitectura del processador mitjançant l'assignació de l'etiqueta “export ARCH”. En el procés de la iteració base l'arquitectura escollida ha estat la del processador del computador on s'ha desenvolupat la distribució: “i386”.

Un cop explicat l'arxiu de configuració, cal explicar el contingut dels arxius associats als perfils. En l'entorn de treball de l'aplicació s'hi troba la carpeta “profiles” on s'han de generar els arxius associats als perfils. El nom d'aquests arxius consisteix en el nom del perfil i una extensió concreta per determinar la tasca d'aquest arxiu. L'extensió “.packages” s'utilitza per determinar quins paquets formaran part de la distribució. La sintaxis d'aquest arxiu consisteix en una llista de paquets separats per salts de línia. L'extensió “.postinst” consisteix en un *script* en *bash* que s'executarà en la fase final de l'instal·lació. Finalment l'extensió “.preseed” s'utilitza per sembrar respostes automàtiques a l'instal·lador. La sintaxis

d'aquest arxiu consisteix en opcions del instal·lador de Debian separades per salts de línia, així, per exemple, es podrà definir el nom del equip de manera automàtica i, si es vulgues, es podria automatitzar tota la instal·lació. Tanmateix automatitzant tota l'instal·lació es perdria flexibilitat i, per tant, cal establir un criteri alhora de sembrar respostes automàtiques.

En el procés de desenvolupament de la iteració base els arxius amb extensió “postinst” i “preseed” no són necessaris i per tant només s'ha de definir el contingut de l'arxiu “default.packages”.

Mitjançant la comanda “dpkg --get-selections | grep -v linux-image | cut -f1” es poden llistar tots els paquets instal·lats en la distribució que s'està utilitzant i, d'aquesta manera, executant aquesta comanda en una distribució Debian acabada d'instal·lar es llisten els paquets necessaris a incloure a “default.packages” i, així també, a la distribució a generar durant la iteració base d'aquesta etapa.

Per finalitzar la primera iteració, només resta, després d'haver creat els arxius necessaris, utilitzar l'aplicació simple-cdd per generar l'instal·lador de la distribució personalitzada Debian base i, finalment, realitzar la fase de proves d'aquesta.

La fase de proves de cadascuna de les iteracions consisteix en la instal·lació i execució del producte resultant de cadascuna de les iteracions en una màquina virtual de l'aplicació VirtualBox. Mitjançant el comportament de la distribució es poden fer petits canvis en el disseny de les següents iteracions de la distribució amb l'objectiu de corregir errors trobats o simplement afegir funcionalitats que resten per resoldre necessitats funcionals.

Un cop realitzada la fase de test de la iteració base comença el breu procés de disseny de la següent iteració.

La resta d'iteracions de la primera etapa

En aquesta segona iteració l'objectiu consistia en deixar enrere Debian i apropar-se a dEICbian mitjançant la satisfacció d'alguns dels requisits funcionals del projecte.

A part d'afegir funcionalitats també cal retallar-ne. La distribució Debian inclou programari no necessari i, que per tant, no forma part dels requisits del projecte.

Tal com es pot veure a la figura 4.2 aquest procés es va anar duent a terme de manera progressiva al llarg de diferents iteracions. És durant la implementa-

ció d'aquestes iteracions quan s'arriba a la conclusió que no es poden satisfer tots els requisits funcionals del projecte mitjançant paquets i, que per tant, cal realitzar *scripts* que instal·lin i configurin aplicacions un cop acabada l'instal·lació de paquets.

Es dissenya, aleshores, un *script* de postinstal·lació amb l'objectiu de solucionar aquestes mancances i, de passada, personalitzar detalls de la distribució. El diagrama de flux dissenyat per la realització de l'*script* es pot veure a la figura 4.4, cal tenir en compte, però, que tot i que s'executarà seqüencialment moltes de les tasques d'aquest són independents i l'ordre en que es facin no afecta en res als resultats.

Per tal d'implementar l'*script* dissenyat cal modificar l'arxiu de configuració i realitzar un nou arxiu amb el propi *script*.

Per tal d'afegir arxius propis com ara el fons de pantalla o l'arxiu de marcadors predefinits amb enllaços útils del navegador web al disc de la distribució cal afegir a l'arxiu de configuració aquesta opció. Per tal de fer-ho s'ha d'afegir a l'arxiu de configuració una línia amb la següent sintaxis:

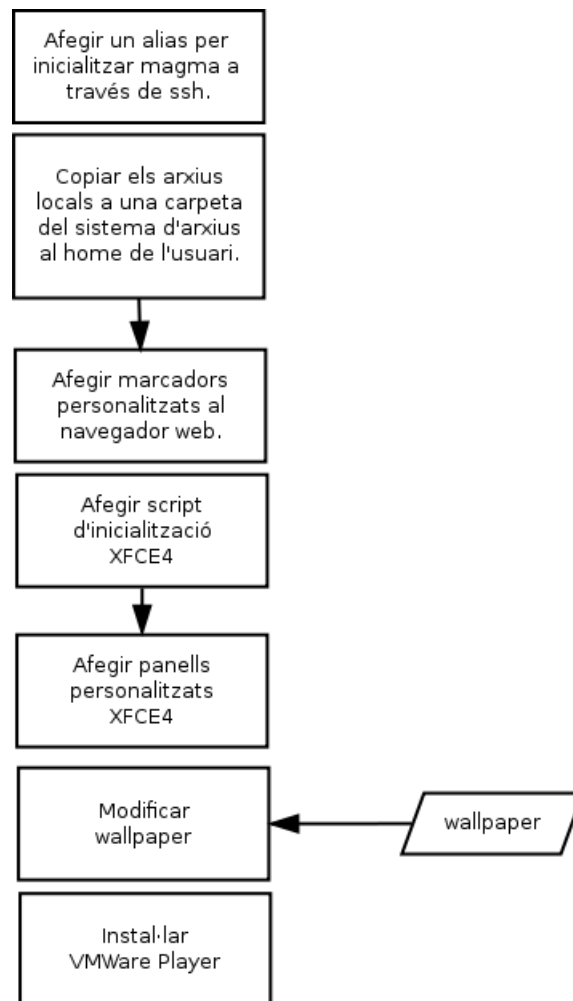
```
all_extras="path_arxiu_1 path_arxiu_2 ... path_arxiu_n"
```

Els arxius afegits al disc d'aquesta manera es trobaran a la carpeta Simple-cdd d'aquest.

Un cop desenvolupat l'*script* de postinstal·lació i afegida la llista de paquets definitiva s'acaba la primera fase i s'obté com a resultat l'instal·lador de la distribució dEICbian.

4.3.2 Generació de Live dEICbian

Una vegada s'ha realitzat l'instal·lador de la distribució, comença un nou cicle d'iteracions que té per objectiu el desenvolupament de Live dEICbian. Aquest procés requereix una nova aplicació anomenada Live Helper [8] que ja s'ha presentat al capítol dos de la memòria. Aquesta secció, tal com l'anterior, ha de servir de documentació bàsica de cara a la continuïtat del projecte i, per tant, es farà una explicació detallada de les etapes crítiques d'implementació.

Figura 4.4: Diagrama de flux del *script* de postinstal·lació

La iteració base

Com en el procés iteratiu anterior, cal definir els objectius de la primera iteració per iniciar d'aquesta manera el cicle de disseny, desenvolupament i test. Els objectius de la iteració base, en aquesta ocasió, són els de realitzar una imatge mitjançant Live Helper amb la configuració i funcionalitats associades per defecte, tot i que fent servir les comandes de baix nivell de l'aplicació. Aquesta primera iteració té com objectiu conèixer les comandes de baix nivell de Live Helper mitjançant el seu ús.

La primera comanda que es crida és “lh config”, que s'encarrega de configurar l'entorn de treball mitjançant la creació d'una carpeta anomenada “config”, on es generen els arxius de configuració necessaris. Per tal que els arxius de configuració siguin apropiats per la generació de la distribució, cal o bé mitjançant l'ús de paràmetres a la comanda o bé l'edició manual dels arxius adequar-los a les necessitats del sistema. En aquests arxius s'han de definir els miralls que farà servir el gestor de paquets, el gestor de paquets, l'arquitectura del processador per la que es genera la distribució, el gestor d'arrancada i altres detalls menors també configurables. Per desenvolupar la iteració base es mantenen les configuracions per defecte i, per tant, no cal editar els arxius de configuració ni associar paràmetres a la crida de la funció.

Un cop s'han creat els arxius de configuració i s'han emplenat correctament es pot iniciar el procés de creació del sistema que es trobarà al iniciar el Live de la distribució. El primer pas és la crida de la comanda “lh bootstrap”. Aquesta comanda crea l'estructura de fitxers i els arxius bàsics d'un sistema Debian per, a partir d'aquest, començar a afegir-hi tot els paquets i configuracions necessaris per definir el sistema d'arxius del Live. Acabat el procés de *bootstrap* i mitjançant l'ús de la comanda “lh chroot” es crearà una carpeta “chroot” sobre la que es treballarà per crear l'entorn de la distribució. Sobre aquesta, i gràcies a la comanda “chroot”, s'instal·larien els paquets necessaris i es farien les modificacions al sistema necessàries. En aquest cas, l'iteració base, aquest procés s'omet.

Amb aquest procés finalitzat només resta generar la imatge ISO on es trobarà el Live de la distribució llesta per ser arrancada. El procés de generar la imatge s'iniciarà mitjançant la comanda “lh binary” que, en primer lloc, comprimirà el sistema de fitxers contingut a la carpeta “chroot” mitjançant “squashfs” per tal que

aquest formi part del Live. Un cop acabat aquest procés es genera la imatge ISO.

La resta d'iteracions de l'etapa

L'objectiu de les següents iteracions consistirà en realitzar un sistema Live amb les mateixes personalitzacions i funcionalitats que l'obtingut mitjançant l'instal·lador de dEICbian. Aquest procés de generació de Live dEICbian consisteix en instal·lar i configurar tots els paquets decidits i realitzar els mateixos passos que es realitzen mitjançant l'*script* de postinstal·lació i, es portaran a terme tal com descriu la figura 4.2.

En primer lloc, però, s'ha de configurar correctament l'aplicación. Tot i que es podria fer mitjançant el pas de paràmetres i la comanda “lh config” es decideix editar manualment els arxius generats per aquesta en l'anterior iteració tot seguint les indicacions dels propis arxius a modificar.

Un cop realitzat aquest pas es continua el procés de generació de Live dEICbian amb la comanda “lh chroot”. Mitjançant l'ús del gestor de paquets apt i la comanda chroot s'instal·len al sistema, contingut a la carpeta “chroot”, tots els paquets necessaris. Un cop instal·lats tots els paquets, i també mitjançant la comanda “chroot”, s'executarà l'*script* de postinstal·lació per tal de modificar el sistema de fitxers del Live de igual manera que es fa durant l'instal·lació.

Un cop realitzades aquestes modificacions i generada la imatge mitjançant la comanda “lh binary”, s'obté com a resultat la imatge ISO de Live dEICbian.

4.3.3 Generació de dEICbian i del generador de distribucions dEICbian

La darrera fase del desenvolupament de dEICbian correspon a la de combinar, en un sol producte, l'instal·lador de dEICbian i Live dEICbian. Per tal de fer-ho, s'ha decidit afegir al Live de la distribució l'instal·lador mitjançant la personalització del carregador d'arrancada GRUB on s'ha afegit una opció per tal d'inicialitzar aquest. No obstant, no només s'ha de modificar el contingut del carregador d'arrancada, sinó que, també, s'han d'afegir els arxius de l'instal·lador dins de la imatge de Live dEICbian. Per tal de realitzar totes aquestes tasques, s'ha continuat els processos iteratius anteriors on el resultat dels quals era precisament les imatges de l'instal·lador i de Live dEICbian.

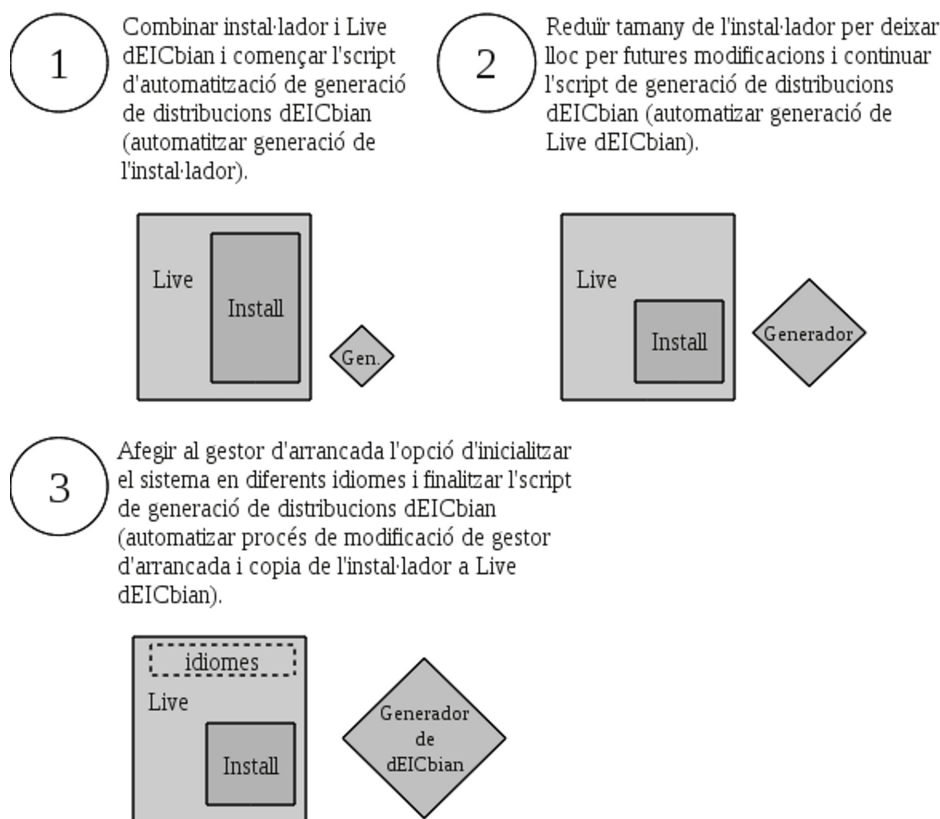


Figura 4.5: Iteracions de la darrera etapa

La iteració base

La primera iteració d'aquesta nova etapa té per objectiu combinar els dos elements que formen dEICbian, de manera que tots dos siguin funcionals i, en posteriors iteracions, ja s'optimitzaran perquè s'utilitzin els recursos dels que es disposa de manera apropiada. També durant aquesta etapa s'inicialitzarà el procés de disseny i implementació del generador de distribucions dEICbian. En resum, es poden veure l'objectiu de les diferents iteracions d'aquesta darrera etapa del desenvolupament a la figura 4.5 i, de forma més general, a la figura 4.2.

En primer lloc s'ha d'accedir a modificar el gestor d'arrancada carregant els mòduls necessaris per inicialitzar el instal·lador. El carregador d'arrancada de l'instal·lador està realitzat amb SYSLinux i, mitjançant l'estudi d'aquest, s'afegeixen les opcions necessàries al carregador d'arrancada GRUB de Live dEICbian.

Un cop afegides les modificacions al carregador d'arrancada cal copiar l'ins-

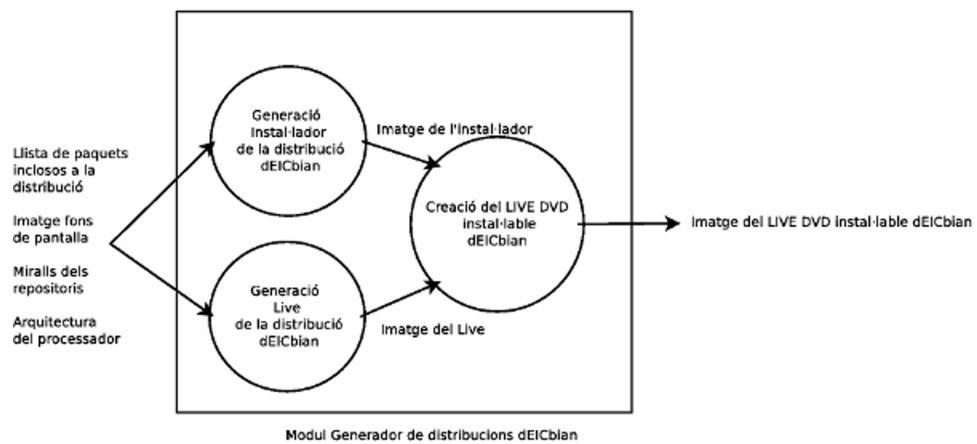


Figura 4.6: Moduls generador distribucions

tal·lador dins de Live dEiCbian i realitzar els enllaços simbòlics adequats per tal que l'instal·lador trobi els arxius necessaris durant el procés d'instal·lació.

Finalitzats aquests dos passos, s'ha de tornar a generar l'imatge ISO però, donat que no s'han fet modificacions al sistema de fitxer comprés mitjançant “squashfs”, no cal executar la comanda “lh binary” perquè mitjançant una comanda de més baix nivell es pot generar la imatge sense tornar a comprimir el sistema de fitxers, aquesta comanda és “lh binary_iso -force”.

Alhora que es realitza aquest procés comença el procés d'implementació del generador de distribucions dEiCbian. El generador consta de tres mòduls diferents que componen cadascuna de les etapes de disseny de la distribució: instal·lador, Live i combinació dels dos elements. El disseny de la interacció entre els mòduls es troba a la figura 4.6.

En aquesta primera iteració és té per objectiu començar el generador i finalitzar l'automatització de la generació de l'instal·lador. Les tasques a realitzar de manera seqüencial pel generador es dissenyen mitjançant un diagrama de flux que es pot veure a la figura 4.7.

En aquesta primera iteració sorgeix una problemàtica. La distribució amb l'instal·lador pràcticament ocupa un DVD sencer de manera que deixa poc espai per futures modificacions quan s'utilitzi el generador de distribucions.

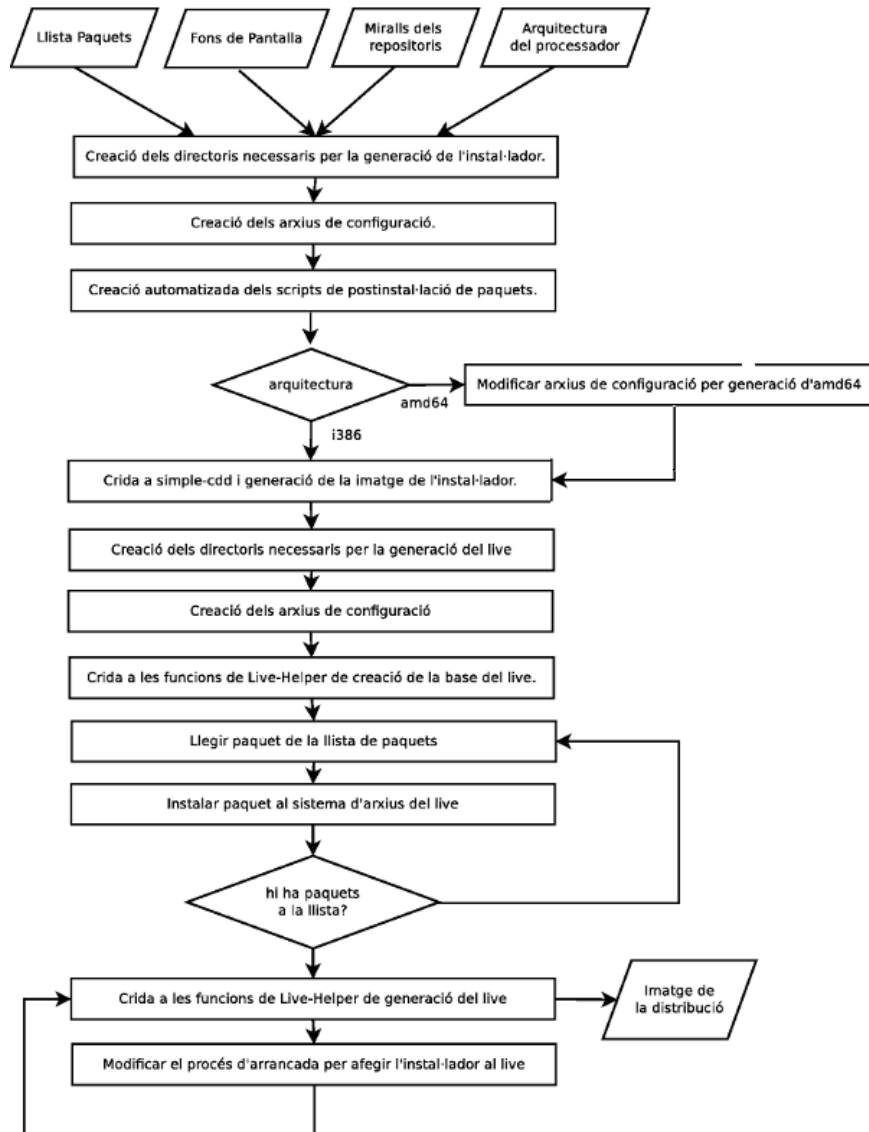


Figura 4.7: Diagrama de flux del generador de noves versions de dEiCbian

Segona iteració

Per tal de solucionar la problemàtica trobada al finalitzar la primera iteració, es decideix, durant el disseny de la segona iteració, disminuir el pes de l'instal·lador i, donat que en les instal·lacions de Debian és necessari l'ús de Internet, es decideix que en lloc d'incloure tots els paquets que formen dEICbian dins de l'instal·lador aquests s'instal·laran mitjançant el gestor de paquets i, per tant, es descarregaran durant la instal·lació. Aquesta decisió disminueix el pes de l'instal·lador considerablement i, donat que el sistema de fitxers del Live es troba comprès es podran generar noves versions dEICbian mitjançant el generador de distribucions dEICbian sense tenir problemes d'espai.

També, durant el disseny d'aquesta segona iteració, i amb motiu de la problemàtica trobada, es dissenya una possible modificació en la distribució que consisteix en aprofitar el sistema de fitxers del Live per realitzar la instal·lació. Aquest disseny es desestima donat que treu moltes opcions de configuració de la instal·lació i, tot i que a nivell d'ús personal de l'alumne podria ser adequat, es troba convenient mantenir l'opció d'instal·lar mitjançant instal·lador, i no pas copia, als laboratoris de docència del departament.

Per tant, després de dissenyar la iteració, cal modificar l'automatització de la generació de l'instal·lador per tal que els paquets es descarreguin i instal·lin mitjançant el gestor de paquets i realitzar el segon mòdul del generador de distribucions dEICbian. A més a més es tornarà a generar una distribució dEICbian amb les modificacions afegides per tal de tornar a realitzar la fase de test d'aquesta.

Com s'ha dit, els objectius de la segona iteració són realitzar les modificacions dissenyades i continuar el desenvolupament del generador conforme a la planificació de les iteracions de la figura 4.2. En resum, s'ha de finalitzar l'automatització de "Live dEICbian" i tornar a realitzar la distribució dEICbian amb l'instal·lador de mida reduïda.

La generació de dEICbian amb l'instal·lador de menor mida no comporta cap problema i es realitza de manera idèntica a la detallada amb anterioritat durant la primera etapa d'iteracions. Pel que fa la generació del *script* d'automatització de "Live dEICbian" es segueix el diagrama de flux de la figura 4.7 comentat anteriorment.

Per finalitzar aquesta iteració es realitza la fase de test, on no es troben proble-

mes ni cap altre detall ressenyable, i es comença, acte seguit, la darrera iteració del desenvolupament.

Darrera iteració

Finalment en la tercera iteració d'aquesta etapa es finalitza el generador i es realitza la distribució dEICbian final mitjançant l'ús d'aquest. Per tal de finalitzar el generador només resta afegir a aquest els passos que s'han realitzat de manera manual durant la primera iteració d'aquesta fase, és a dir, modificar el gestor d'arrancada per incloure l'instal·lador i copiar l'estructura de fitxers d'aquest dintre de la imatge de "Live dEICbian". Finalment, durant la modificació automàtica del gestor d'arrancada s'ha inclòs el procés d'arrancada en tres idiomes diferents: Català, Castellà i Anglès, per tal d'adaptar-se als diferents perfils d'alumnes (Erasmus, Séneca, locals).

Un cop s'han introduït les modificacions del gestor d'arrancada a l'*script* s'acaba el generador de noves versions de dEICbian.

Amb el generador acabat es genera, utilitzant aquest, la versió definitiva de dEICbian. Es realitza, abans de donar per acabat el desenvolupament, la fase de test de la iteració en la que s'instal·la i es prova la distribució. Un cop comprovat que tot funciona correctament es dona per acabat el procés de desenvolupament iteratiu.

Acabada aquesta última etapa del desenvolupament tenim com a resultat la anomenada distribució dEICbian i el generador de noves versions de la distribució.

4.3.4 Realització de documentació d'ajuda

De forma conjunta a la distribució s'inclou en el projecte la realització d'una pàgina web que albergarà la documentació i la imatge de la distribució dEICbian.

La documentació s'ha realitzat un cop finalitzada la distribució dEICbian i, per tant, fora del cicle de vida iteratiu del projecte.

En la documentació s'hi pot trobar el programari principal que forma part de la distribució, una sèrie de preguntes freqüents resoltes i una petita guia d'instal·lació de la distribució.

La pàgina de documentació de la distribució formarà part de la pàgina web del departament d'enginyeria de la informació i de les comunicacions.

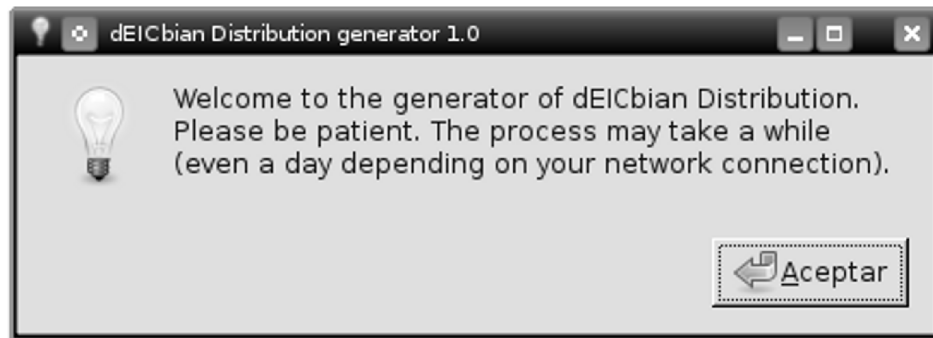


Figura 4.8: Diàleg de benvinguda del generador de distribucions

4.3.5 Interfície i ús del generador

La interfície del generador consisteix en crides a la aplicació Zenity [9] comentat al segon capítol de la memòria. En aquesta secció de la memòria es detalla quin ha estat el seu ús en la realització de la interfície del generador tot detallant les possibilitats que té l'usuari alhora de generar una distribució.

En primer lloc el generador dóna un missatge de benvinguda i adverteix que el procés de generació de la distribució pot trigar força temps amb el diàleg que es pot veure a la figura 4.8.

Tot seguit comença el procés de selecció de paràmetres del generador que es realitza mitjançant un seguit de diàlegs d'entrada. En primer lloc es selecciona la llista de paquets, el fons de pantalla de la distribució, els miralls i, finalment, l'arquitectura del processador. A la figura 4.9 es pot veure el diàleg de selecció de la llista de paquets.

Un cop el generador de distribucions acaba la generació de la nova versió de la distribució dEICbian s'avis a l'usuari mitjançant el diàleg de la figura 4.10 tot informant d'on es troba la imatge ISO de la distribució.

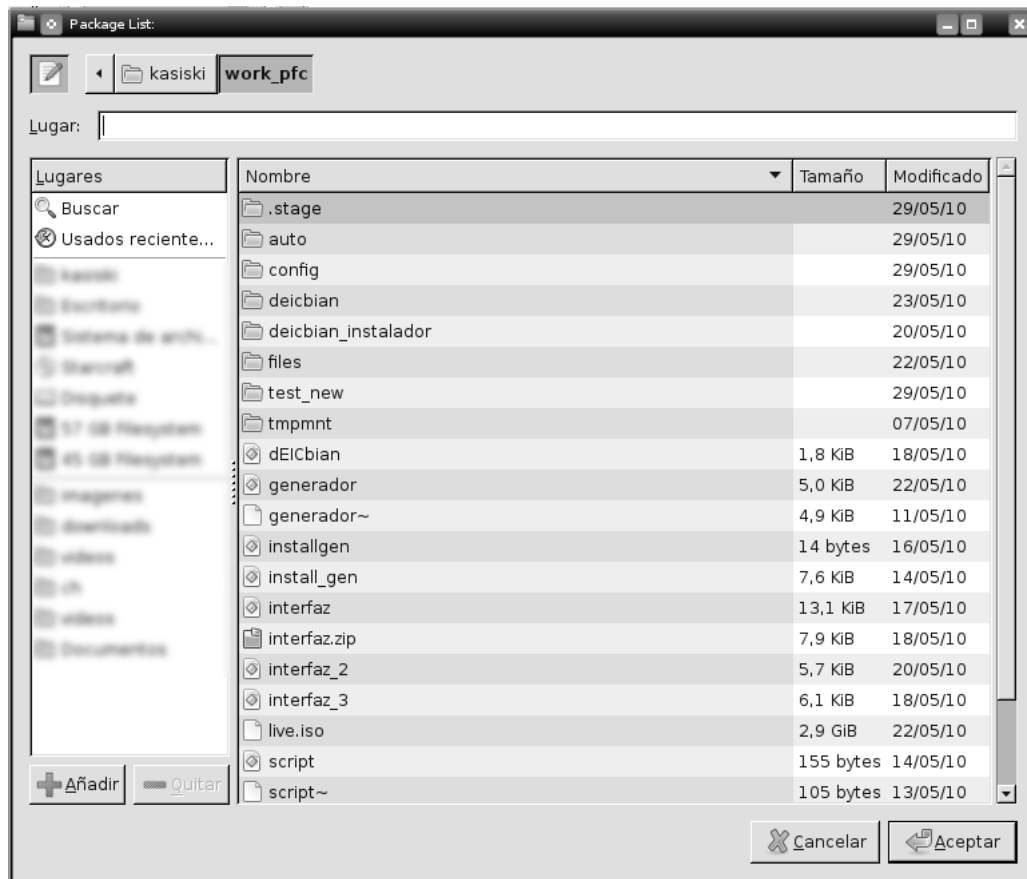


Figura 4.9: Diàleg de selecció de la llista de paquets a instal·lar

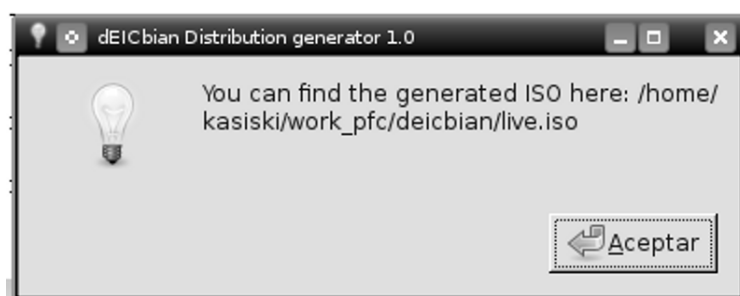


Figura 4.10: Diàleg final del generador

Capítol 5

Planificació i costos

A l'inici del projecte es va realitzar una planificació temporal i econòmica que es va presentar a l'informe previ d'aquest. En aquest capítol de la memòria es presenta una comparació entre la planificació original i el desenvolupament real del projecte. També es tindrà en compte quines de les tasques inicialment planificades s'han realitzat i quines modificacions han sofert.

5.1 Tasques i planificació temporal

Es detallen a continuació les tasques realitzades durant l'elaboració del projecte, indicant les fites entre claudàtors. Les tasques principals s'indiquen mitjançant l'ús de negreta. A més a més es detallen quines tasques principals no s'havien planificat inicialment mitjançant l'ús de la font en cursiva.

- (1)**Anàlisi de requisits [F1]**

- (1.1) Anàlisi de programari necessari de la Distribució

- (1.1.1) Assignatures que es duen a terme als laboratoris de docència

- (1.1.2) Professors que imparteixen les assignatures

- (1.1.3) Previsió de programari necessari per les assignatures

- (1.1.2) Entrevistes professors pràctiques

- (1.1.3) Cerca d'alternatives per al programari privatiu del que no es disposen llicències

(1.1.4) Elaboració de la llista definitiva de programari que ha d'estar inclòs a la Distribució

(1.2) Anàlisi de requisits no funcionals

- **(2) Creació d'un instal·lador de la distribució base personalitzada Debian [F2]** (CBD - Custom Base Distribution) [iteració base]

(2.1) Instal·lació del programari necessari per la creació de distribucions Debian

(2.2) Creació d'una distribució amb programari bàsic

(2.3) Test de la Distribució

- **(3) Estudi de les aplicacions i dels paquets a instal·lar**

(3.1) Estudi de les aplicacions per al desenvolupament de distribucions

(3.2) Distribuir el programari en dos grups segons l'existència (o no) de paquets

(3.3) Instal·lació i test del programari del primer grup en Debian

(3.4) Afegir el programari del primer grup a la CBD i fer el test de nou de la CBD

(3.5) Creació de *scripts* d'instal·lació del programari del segon grup

(3.6) Instal·lació i test del programari del segon grup en Debian

(3.7) Afegir el programari del segon grup a la CBD i fase de test de la CBD

- **(4) Creació d'una distribució amb tot el programari decidit**

(4.1) Creació de la distribució dEICbian

(4.1.1) Creació de l'instal·lador de la distribució dEICbian [F3] [primera etapa d'iteracions]

(4.1.2) Creació de la versió Live de la distribució dEICbian [F4] [segona etapa d'iteracions]

(4.1.3) Combinar l'instal·lador i la versió Live per crear dEICbian [F5] [tercera etapa d'iteracions]

(4.2) Test final de la distribució - Anàlisi del resultat

- (5) **Creació d'un generador de la distribució [F6]**
 - (5.1) Creació dels *scripts* del generador
 - (5.2) Ús del generador per realitzar la distribució.
- (6) **Realització de la memòria i preparació de la presentació [F7]**
 - (6.1) Redacció de la memòria
 - (6.2) Preparació de la presentació

A continuació, a la figura 5.1, es pot veure una comparació entre el diagrama de Gantt planificat inicialment amb el que ha resultat en realitat. Per tal de reduir l'extensió dels diagrames aquests estan resumits tenint en compte únicament les tasques principals i les fites de desenvolupament.

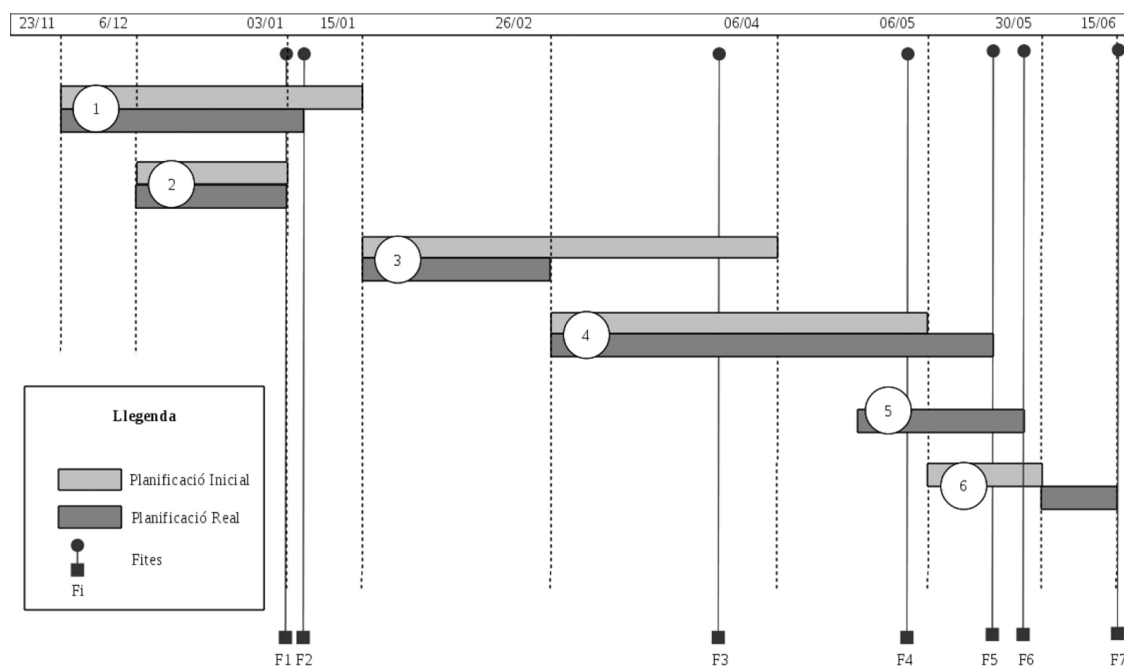


Figura 5.1: Planificació temporal

5.2 Costos

A la taula 5.1 es pot veure una comparació entre el cost inicialment planificat del projecte amb el cost estimat resultant del desenvolupament d'aquest.

Per tal d'efectuar la taula de costos s'ha establert un sou de 26 Euros/hora corresponent al sou d'un Tècnic de Suport a la Recerca del capítol VI de la UAB.

Cal destacar que, respecte la planificació inicial, hi ha un decrement de cost important que es deu principalment al fet d'haver reduït considerablement el temps d'anàlisi de requisits, entre altres coses, per haver realitzat les entrevistes amb els docents de les sessions de laboratori mitjançant correu electrònic.

També es va fer una previsió massa alta el treball d'estudi del programari donat que, inicialment, no es coneixia quin seria i podrien haver sorgit problemes. A la pràctica no ha estat així, i per tant, s'ha realitzat en menys temps del planificat.

Finalment també s'ha disminuït el temps de test de la distribució gràcies a la col·laboració d'alumnes d'últim any d'Enginyeria Informàtica que, mitjançant la prova de la distribució al seu ordinador personal i una enquesta de satisfacció, han facilitat la realització d'aquesta etapa.

5.3 Resum

Es pot observar en aquest capítol una comparació entre la planificació temporal inicial i els costos associats i les variacions que aquesta planificació ha sofert al llarg del projecte. Per tal de donar aquesta informació es pot trobar un llistat de tasques i subtasques realitzades, un diagrama de Gannt amb les tasques i fites principals i, finalment, una taula comparativa de temps i cost entre la planificació inicial i la real.

Tasca	Hores previstes	Hores reals	Δ hores	Cost previst	Cost real	Δ cost
1	76	26	-50	1976	676	-1300
1.1	36	14	-22	936	364	-572
1.2	16	4	-12	416	104	-312
1.3	24	8	-16	624	208	-416
2	24	24	0	624	624	0
2.1	4	4	0	104	104	0
2.2	8	8	0	208	208	0
2.3	12	12	0	312	312	0
3	140	50	-90	3640	1300	-2340
3.1	0	2	2	0	52	52
3.2	4	2	-2	104	52	-52
3.3	16	12	-4	416	312	-104
3.4	12	4	-8	312	104	-208
3.5	36	12	-24	936	312	-624
3.6	24	6	-18	624	156	-468
3.7	48	12	-36	1248	312	-936
4	120	96	-24	3120	2496	-624
4.1	30	72	42	780	1872	1092
4.2	90	24	-66	2340	624	-1716
5	0	36	36	0	936	936
5.1	0	24	24	0	624	624
5.2	0	12	12	0	312	312
6	90	75	-15	1080	1950	-180
6.1	60	60	0	1560	1560	0
6.2	30	15	-15	780	390	-180
Totals	450	307	-143	11700	7982	-3718

Taula 5.1: Costos

Capítol 6

Conclusions

En aquest capítol es presenten les conclusions del projecte classificades en tres blocs. El primer bloc consisteix en una valoració detallada de la satisfacció d'objectius del projecte mitjançant l'explicació del producte resultant que s'obté. En segon lloc, es reflexionarà sobre els resultats. Finalment, s'introdueixen possibles línies futures d'actuació sobre el projecte realitzat.

6.1 Valoració

Resumint la situació inicial expressada a la introducció es mira d'explicar en aquest punt l'assoliment dels objectius del projecte.

En primer lloc, trobàvem que els alumnes havien de realitzar sessions pràctiques parcialment fora del laboratori, on es troben amb un sistema diferent amb el que havien iniciat la pràctica i en el que s'avaluarà aquesta. Mitjançant la distribució dEICbian, aquesta problemàtica queda solucionada perquè, mitjançant Live dEICbian, els alumnes poden tenir fàcilment el mateix sistema que es troben als laboratoris. Si bé es cert que fins que no es posi en funcionament aquest sistema no es podrà tenir certesa de la seva acceptació i usabilitat s'han realitzat enquestes de valoració satisfactòries. Els resultats d'aquestes enquestes es trobarà a la següent secció de la memòria, dedicada als resultats.

En segon lloc vèiem que si el sistema dels ordinadors del laboratori quedava malmès, aquest s'havia de reinstal·lar, de manera que el programari també s'hauria de tornar a instal·lar i configurar. Mitjançant dEICbian hi ha dos possibles so-

lucions: L'ús de Live dEICbian per evitar els processos de instal·lació o bé l'ús de l'instal·lador del sistema que instal·larà i configurarà alhora tot el programari necessari.

Es cert que, especialment durant el primer any de funcionament del sistema, probablement sorgeixin noves necessitats funcionals que haurien de quedar solucionades mitjançant el generador de distribucions dEICbian, que facilita la generació de noves versions de la distribució.

A continuació, es detalla l'assoliment dels objectius concrets del projecte, comentats a la introducció de la memòria. L'estudi de les distribucions personalitzades, així com el de les eines per al desenvolupament, s'ha dut a terme mitjançant l'anàlisi de l'estat de l'art i durant la generació de la distribució personalitzada. El tercer objectiu concret consistia en trobar les necessitats dels alumnes i membres del departament que s'ha tractat al llarg de l'anàlisi de requisits. Els punts referits al disseny i la implementació de la distribució s'han explicat acuradament al quart capítol de la memòria. Finalment, l'últim objectiu concret feia esment de la validació de la distribució que es discuteix al següent apartat de la memòria destinat a comentar els resultats.

6.2 Resultats

Per afrontar la validació dels resultats, s'han realitzat una sèrie d'enquestes a un conjunt d'alumnes de darrer any d'enginyeria informàtica, que han realitzat al llarg de la carrera diverses de les sessions pràctiques de les assignatures del departament. Es considera que la seva opinió es de gran valor perquè han afrontat i superat la realització de les sessions de laboratori en les que han participat i, d'aquesta manera, seran capaços de valorar si consideren apropiada la distribució per aquest ús. L'enquesta consisteix en un seguit de qüestions valorables numèricament amb quatre valors amb el següent significat:

- 1 - No.
- 2 - Potser (Més aviat no).
- 3 - Possiblement (Més aviat sí).
- 4 - Sí.

Les preguntes realitzades són les següents:

1. Vas trobar dificultats de configuració d'aplicacions al realitzar pràctiques de assignatures del dEIC?
2. Creus que la distribució ajudaria a evitar aquests problemes de configuració d'aplicacions?
3. Instal·laries la distribució al teu ordinador personal si haguessis de realitzar les pràctiques d'alguna assignatura del dEIC?
4. Utilitzaries la versió Live de la distribució al teu ordinador personal si haguessis de realitzar les pràctiques d'alguna assignatura del dEIC?
5. Consideres apropiada la distribució pel seu ús als laboratoris del dEIC?
6. Consideres el ventall de programari prou ampli (i suficient) com per realitzar les pràctiques del dEIC?
7. Consideres adequada la solució d'executar "magma" de manera remota per tal de solucionar la seva mancança (deguda a termes legals)?

A la figura 6.1 es poden veure els resultats de l'enquesta. En primer lloc, cal destacar els bons resultats de satisfacció de la distribució que, en general, es considera apropiada pel seu ús en docència. D'altra banda, sembla que pocs estudiants estaran disposats a instal·lar la distribució al seu ordinador personal si bé es cert que consideren favorable la idea d'utilitzar la versió Live de la distribució puntualment per tal de realitzar les pràctiques. Aquest fet confirma que ha estat un disseny apropiat, tal com es preveia durant l'anàlisi de requisits la distribució que havia de ser utilitzable fàcilment sense instal·lacions ni configuracions costoses. La creació d'una versió Live sembla solucionar aquest problema.

En resum, les sis primeres preguntes de l'enquesta, destinades a conèixer l'opinió global dels usuaris sobre la distribució i la seva motivació han confirmat el plantejament inicial i han validat favorablement la distribució.

Finalment, la darrera pregunta de l'enquesta, força més concreta, està destinada a conèixer la opinió del usuari sobre la decisió de disseny presa per solucionar la

# Pregunta	Resposta Mitja	Resposta més freqüent	Respostes que s'han donat
1	2,77	3	Totes
2	3,77	4	3 i 4
3	1,44	1	1,2 i 3
4	3	3	2,3 i 4
5	3,66	4	3 i 4
6	3,55	3 i 4	3 i 4
7	3,28	4	3 i 4 (En blanc)

Taula 6.1: Resum de respostes

manca del programari “magma” necessari per realitzar les pràctiques de teoria de la codificació. En general la solució de la connexió de manera remota mitjançant terminal ha estat ben acollida tot i que, cal tenir en compte, que no tots els enquestats han realitzat teoria de la codificació i, els que ho han fet, pot ser que trobin adequada la solució perquè, abans, ja ho havien de fer així.

6.3 Línies de continuïtat

En aquest projecte les línies de continuïtat es poden classificar en dos vessants: La continuïtat de la distribució mitjançant la generació de noves versions i la continuïtat del generador.

Respecte la primera vessant, la continuïtat de la distribució, s'ha de considerar l'actualització del sistema conforme aquesta quedi obsoleta. També hi ha aspectes a millorar que donada la naturalesa del projecte no s'han pogut afrontar en la seva totalitat com ara l'ús exclusiu de programari lliure en les sessions pràctiques permetria grans avanços en la distribució. Altres aspectes a millorar són la personalització total de la distribució donant-li un segell més distintiu que, tot i no ser un aspecte fonamental, podria incrementar el nombre d'alumnes disposats a instal·lar el sistema.

És respecte la segona vessant, la continuïtat del generador, on hi ha línies de continuïtat més ambicioses i on es centra el gruix de possibles millores. En primer lloc la realització del *script* en *bash* té certes limitacions alhora de realitzar interfícies gràfiques i, tot i que s'ha solucionat mitjançant l'ús de diàlegs, millorar aquest aspecte seria força significatiu. La implementació del generador en llenguatges

de programació que permetin incloure una interfície gràfica potent seria el primer gran pas alhora de tenir un generador de noves versions senzill i potent.

D'altra banda, encara centrats en el generador, es podria millorar lleugerament la velocitat del generador, especialment en el cas que es generin noves versions de la distribució de manera molt continuada, si aquest inclogués la generació d'un mirall propi de la distribució base que permetria que la descarrega de paquets es convertís en una còpia de fitxers dins del mateix sistema disminuint el temps d'aquest procés considerablement.

Bibliografia

- [1] The GNU Operating System, juny 2010.
<<http://www.gnu.org/>>
- [2] The Linux Home Page at Linux, juny 2010.
<<http://www.linux.org/>>
- [3] Anatomy of the Linux kernel, juny 2010.
<<http://www.ibm.com/developerworks/linux/library/l-linux-kernel/>>
- [4] Linux Boot Loaders Compared, juny 2010.
<<http://www.xs4all.nl/~lennartb/bootloaders/>>
- [5] Estructura del Sistema de Archivos de Linux, juny 2010.
<<http://es.tldp.org/Estandares/fsstnd12.html/>>
- [6] Filesystem Hierarchy Standard, juny 2010.
<<http://www.pathname.com/fhs/pub/fhs-2.3.pdf/>>
- [7] Simple-CDD - Debian Wiki, juny 2010.
<<http://wiki.debian.org/Simple-CDD/>>
- [8] DebianLive - Debian Wiki, juny 2010.
<<http://wiki.debian.org/DebianLive/>>
- [9] Manual de Zenity, juny 2010.
<<http://library.gnome.org/users/zenity/stable/>>
- [10] Operating System share for 06/2010 | TOP500 Supercomputing Sites, juny 2010.
<<http://www.top500.org/stats/list/35/os>>

- [11] IBM is Committed to Linux and Open Source, juny 2010.
<ftp://ftp.software.ibm.com/linux/pdfs/IBM_and_Linux.pdf>
- [12] HP Open Source and Linux, juny 2010.
<<http://h71028.www7.hp.com/enterprise/cache/309906-0-0-0-121.html>>
- [13] Linux Timeline, juny 2010.
<<http://www.linuxjournal.com/article/9065>>
- [14] Estimating Linux's Size, juny 2010.
<<http://www.dwheeler.com/sloc/redhat62-v1/redhat62sloc.html>>
- [15] Inside the Linux boot process, juny 2010.
<<http://www.ibm.com/developerworks/linux/library/l-linuxboot/>>
- [16] VirtualBox, juny 2010.
<<http://www.virtualbox.org/>>
- [17] VmWare - Introducción a la virtualización, máquina virtual, juny 2010.
<<http://www.vmware.com/es/overview/>>
- [18] About Qemu, juny 2010.
<<http://qemu.org>>
- [19] What is iterative development?, juny 2010.
<<http://www.ibm.com/developerworks/rational/library/may05/bittner-spence/>>
- [20] Réplicas de Debian en todo el mundo, juny 2010.
<<http://www.debian.org/mirror/list>>

Firmat: Antonio Rota Moreno
Bellaterra, Juny de 2010

Resum

En aquest projecte es desenvolupa una distribució GNU/Linux adaptada als laboratoris de docència del departament d'enginyeria de la informació i de les comunicacions. Partint de un anàlisi exhaustiu de les necessitats del departament i gràcies a la col·laboració dels seus membres s'ha realitzat la distribució dEICbian, un sistema que permetrà facilitar les tasques docents als laboratoris del departament. Amb l'objectiu de mantenir el sistema actualitzat també s'ha desenvolupat un generador de noves versions de la distribució dEICbian per tal de garantir tenir, sempre, un sistema modern i lliure d'errors als laboratoris del departament. L'opinió dels usuaris, expressada mitjançant enquestes, ha estat favorable i ara només resta el darrer pas: la seva implementació als laboratoris.

Resumen

En este proyecto se desarrolla una distribución GNU/Linux adaptada a los laboratorios de docencia del departamento de ingeniería de la información y de las comunicaciones. Partiendo de un análisis exhaustivo de las necesidades del departamento y gracias a la colaboración de sus miembros se ha realizado la distribución dEICbian, un sistema que permitirá facilitar las tareas docentes en los laboratorios del departamento. Con el objetivo de mantener el sistema actualizado se ha desarrollado un generador de nuevas versiones de la distribución dEICbian para garantizar tener, siempre, un sistema moderno y libre de errores en los laboratorios del departamento. La opinión de los usuarios, expresada mediante encuestas, ha sido favorable y ahora sólo queda el último paso: su implementación en los laboratorios.

Abstract

This projects develops a GNU/Linux distribution for the teaching laboratories of the department of information and communication engineering. Starting with an analysis of the needs of the department and with the coloboration of its members, the dEICbian distribution has been made. dEICbian is intended to facilitate the teaching work at the practical sessions of the deparment. In order to maintain the current system has developed a generator of new versions of the distribution to ensure dEICbian provides a modern and error-free system to the department's laboratories. The people opinion, expressed through surveys, has been positive and now only one step remains: its implementation on the laboratories.